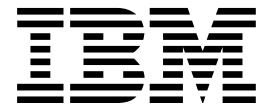


GDDM



Diagnosis

Version 3 Release 2

GDDM



Diagnosis

Version 3 Release 2

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page xiii.

Second Edition (September 1996)

This edition applies to the following IBM GDDM series of licensed programs:

Program number	Program name	Version	Release	Modification
5695-167	GDDM/MVS	3	2	0
5684-168	GDDM/VM	3	2	0
5686-057	GDDM/VSE	3	2	0
5668-812	GDDM-PGF	2	1	3
5668-723	GDDM-IVU	1	1	3
5668-802	GDDM-GKS	1	1	3
5668-801	GDDM Interactive Map Definition	2	1	3

GDDM/MVS as an element of OS/390 (program number 5645-001)

and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Consult the latest edition of the applicable IBM system bibliography for current information on this product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

At the back of this publication is a page titled “Sending your comments to IBM”. If you want to make comments, but the methods described are not available to you, please address them to:

IBM United Kingdom Laboratories, Information Development, Mail Point 095, Hursley Park, Winchester, Hampshire, England, SO21 2JN.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1979, 1996. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	xiii
Trademarks and service marks	xiv
Programming interface information	xv
Preface	xvii
Who this book is for	xvii
What you need to know to understand this book	xvii
How to use this book	xvii
Conventions	xvii
Summary of changes	xix
Changes for GDDM Version 3 Release 2	xix
Changes for GDDM Version 3 Release 1	xix
Chapter 1. Diagnostic procedures	1
Initial diagnosis	1
Unexpected output	1
Device checks and sense codes	2
GDDM error messages	2
Abends	3
Where to look for further advice	3
Problems with application programs	3
Problems with GDDM utility programs	3
Problems with GDDM print utilities	4
Problems with the hardware	4
Problems with customizing workstations	4
Problems with the operating system	4
Problems with GDDM after its installation	5
Problems within GDDM	5
What to do if things go wrong	5
Checking hardware characteristics	6
General printer diagnosis	7
IBM 3287 printer diagnosis	8
IBM 3268 printer diagnosis	9
IBM 3193 display station diagnosis	9
IBM 3274 controller diagnosis	10
IBM 3174 controller diagnosis	11
Non-IBM devices diagnosis	12
Common installation errors and pitfalls	12
Problems associated with abends	13
Problems associated with incorrect output	13
Problems associated with messages	14
Problems involving system performance	18
Problems involving device checks	18
Problems associated with SNA sense codes	19
Problems associated with Graphics Control Program errors	20

Chapter 2. GDDM tracing	21
The TRCESTR external default keyword	22
Starting trace under different subsystems	22
Tracing under TSO	22
Tracing in the TSO print utility environment	23
Tracing under VM	23
Tracing under CICS	23
Tracing under IMS	24
Tracing under VSE/Batch	25
Tracing in an application program	25
Tracing multiple instances of GDDM	25
Sample trace statements	26
Trace keywords	27
Coding TRCESTR statements	27
Functions available with the TRCESTR keyword	30
CLEAR	30
FORCE	30
Action functions	30
Event functions	38
Control function	42
The TRCEWID keyword	42
Format of trace output	42
Trace record type	43
Trace explanation	43
Examples of trace output from TRCESTR statements	44
Example 1. Tracing a range of events	44
Example 2. Restricting the level at which tracing takes place	46
Example 3. Requesting a full transmission trace	49
Example 4. Including the contents of registers in your trace	52
Example 5. Tracing calls to the 5080 Graphics System workstation	55
Example 6. Tracing in multiple instances of GDDM	57
Example 7. Trace output containing CDPDS input	58
Example 8. Trace output containing HRIG output	59
Example 9. Trace output containing PostScript output	60
Example 10. Tracing an imported CGM	62
Example 11. Tracing an exported CGM	64
Other tracing methods	69
The FSTRCE statement	70
Parameters	70
Abend dump output	72
Locating control blocks and the in-storage trace table	72
Locating the currently active GDDM module	73
Locating the GDDM control blocks from the current module	73
Locating the in-storage trace table	74
Locating the in-storage trace table if the save area chain is corrupted	75
Format of in-storage trace table	75
Meaning of in-storage trace records	76
Format for CPNIN, CPNOUT	76
Format for SUBIN, SUBOUT, MODIN, MODOUT	77
Chapter 3. IKT0405I debugging procedure	79
Starting the traces	79
Enabling the GDDM trace	79
Starting the Generalized Trace Facility for data stream errors	79

Starting the VTAM trace	80
Recreate the error	80
The dynamic dump	80
Stopping the traces	82
Disable GDDM tracing	82
Stop GTF	82
Stop VTAM tracing	83
Format the trace	83
How to locate the failing component	83
 Chapter 4. GDDM Interactive Map Definition diagnosis	 85
Using the GDDM-IMD diagnostic facility	85
Invoking the GDDM-IMD diagnostic facility	85
The GDDM-IMD diagnostic frame	85
Main storage display	86
Example addresses	86
Tracing MSL operations	87
Using the MSL traces	87
The MSL access trace	88
The MSL record trace	88
Record validation	89
 Chapter 5. GDDM-REXX diagnosis	 91
GDDM-REXX under MVS	92
REXX and GDDM-REXX tracing under TSO	92
GDDM-REXX under VM	93
REXX and GDDM-REXX tracing under VM	93
 Chapter 6. GDDM-PCLK diagnosis	 95
Common errors with GDDM-PCLK	95
Applying service to GDDM-PCLK	98
Finding the level of GDDM-PCLK code	98
GDDM-PCLK trace support	98
Trace points	99
Trace options	100
Trace control keys	101
Trace output	101
Trace message data file	103
Changes to the default image symbol set	103
Debugging Personal Communications/3270 problems	104
Tracing Personal Communications/3270 Communications	104
Reporting Personal Communications/3270 Problems	105
 Chapter 7. GDDM-OS/2 Link diagnosis	 107
Finding the level of OS/2 code	107
Applying service to GDDM-OS/2 Link	108
Finding the level of GDDM-OS/2 Link code	108
GDDM-OS/2 Link trace support	109
Running the GDDM-OS/2 Link Service Trace Utility	109
GDDM-OS/2 Link Service Trace Utility pull-down menus	110
Trace options windows	112
Tracing output	116
Trace as part of the OS/2 Communications Manager	117
Performing a communications trace	117

Obtaining trace data during normal communications operation	117
Obtaining trace data during the OS/2 Communications Manager auto-start	118
Using the Error Log entries from the OS/2 Communications Manager	120
How to interpret the error data in log messages	121
How to complete an OS/2 Problem Report Form	122
When and why you should use this form	122
Problem Report Form	128
Chapter 8. Reporting GDDM problems to IBM	135
Reporting a problem using a network tool	135
Reporting a problem using a problem report form	136
Defining the symptom of the problem	138
Unexpected, incorrect, or missing output	138
Device program check	138
GDDM message	138
GDDM abend	139
System abend	139
Loop	139
Reduced performance	139
Wait	139
GDDM documentation error	140
Defining the operating environment	140
Defining the devices in use	141
Reporting the problem	142
Submitting an APAR	142
Appendix A. Data areas	143
ADMTAAB – application anchor block (AAB)	144
Function	144
Alphabetic index to field names	144
Storage layout	144
ADMTACA – application interface control area	144
Function	144
Alphabetic index to field names	144
Storage layout	145
ADMTQYV – VM display communication block	148
Function	148
Alphabetic index to field names	148
Storage layout	148
ADMTDFT – general defaults table	151
Function	151
Alphabetic index to field names	151
Storage layout	152
ADMTDSA – dynamic save area (DSA)	156
Function	156
Alphabetic index to field names	157
Storage layout	157
ADMTDSAO – dynamic save area overflow stack (DSAO)	158
Function	158
Alphabetic index to field names	158
Storage layout	158
ADMTDSAS – dynamic save area stack (DSAS)	158
Function	158
Alphabetic index to field names	158

Storage layout	158
ADMTIFCB – interface control block (IFCB)	159
Function	159
Alphabetic index to field names	159
Storage layout	160
ADMTIFCE – GDDM interface control error feedback	163
Function	163
Alphabetic index to field names	163
Storage layout	163
ADMTNICK – nickname list entry format	164
Function	164
Alphabetic index to field names	164
Storage layout	164
ADMTTCA – terminal services interface control area (TCA)	165
Function	165
Alphabetic index to field names	165
Storage layout	165
ADMTTDB – terminal descriptor block (TDB)	166
Function	166
Alphabetic index to field names	166
Storage layout	166
ADMTTRB – terminal request block (TRB)	168
Function	168
Alphabetic index to field names	169
Storage layout	169
Appendix B. Abend codes	173
Appendix C. Message-to-module cross-reference	185
GDDM-IMD messages	195
Appendix D. Trace-string syntax	199
Recognized tokens	200
Appendix E. Request Control Parameter codes	203
Glossary	213
Bibliography	229
Latest GDDM information	229
GDDM publications	229
Books from related libraries	230
Index	233

Figures

1.	GDDM-PCLK Main Panel	98
2.	GDDM-PCLK Service Functions	99
3.	GDDM-PCLK Trace	100
4.	GDDM-OS/2: Example of SYSLEVEL output	108
5.	GDDM-OS/2 Link: Update Trace Options window	112
6.	GDDM-OS/2 Link: Environment Trace Options window	113
7.	GDDM-OS/2 Link: Window Procedure Trace Options window	114
8.	GDDM-OS/2 Link: Session n Trace Options window	115
9.	An example of an error log entry full-screen display	121
10.	Example of log message error data	122
11.	Priority Checklist (Part 1)	125
12.	Priority Checklist (Part 2)	126

Tables

1. GDDM Base RCP codes, in numeric order	203
2. GDDM-PGF RCP codes, in numeric order	209

Notices

The following paragraph does not apply to any country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service in this publication is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the responsibility of the user.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact Laboratory Counsel, Mail Point 151, IBM United Kingdom Laboratories, Hursley Park, Winchester, Hampshire SO21 2JN, England. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, U.S.A.

Trademarks and service marks

The following terms, used in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

APL2	CICS	CICS/ESA
CICS/VSE	GDDM	graPHIGS
IBM	IBMLink	MVS/XA
Operating System/2	OS/2	Personal System/2
Presentation Manager	PS/2	VM/ESA
VM/XA	VSE/ESA	VTAM
OS/390		

The following terms, used in this publication, are trademarks of other companies:

DEC	Digital Equipment Corporation
Tektronix	Tektronix Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

C-bus is a trademark of Corollary, Inc.

Microsoft, Windows and the Windows 95 Logo are trademarks or registered trademarks of Microsoft Corporation.

Programming interface information

This book is intended to help you diagnose problems that may occur when you use GDDM.

This book also documents General-use Programming Interface and Associated Guidance Information, Product-sensitive Programming Interface and Associated Guidance Information, and Diagnosis, Modification or Tuning Information provided by GDDM.

General-use programming interfaces allow the customer to write programs that obtain the services of GDDM.

General-use Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

General-use programming interface

General-use Programming Interface and Associated Guidance Information...

End of General-use programming interface

Product-sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of GDDM. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Product-sensitive programming interface

Product-sensitive Programming Interface and Associated Guidance Information...

End of Product-sensitive programming interface

Diagnosis, Modification or Tuning Information is provided to help you diagnose problems with GDDM.

Warning: Do not use this Diagnosis, Modification or Tuning Information as a programming interface.

Diagnosis, Modification or Tuning Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

_____ Diagnosis, Modification or Tuning Information _____

Diagnosis, Modification or Tuning Information...

_____ End of Diagnosis, Modification or Tuning Information _____

Preface

This book is intended to help you diagnose problems that may occur when using the IBM GDDM series of products. It primarily contains problem determination procedures for GDDM running under MVS (and its subsystems CICS, TSO, and IMS), VM, VSE, and as an element of OS/390.

Who this book is for

This book is for system support personnel for GDDM, and for IBM service personnel.

What you need to know to understand this book

This book assumes that you are familiar with GDDM, that you have some knowledge of debugging application and system problems, and that you understand the task that the application program is intended to perform.

If you are not familiar with GDDM, you should read some of the books in the GDDM library, starting with the *GDDM General Information* manual. The books of the GDDM library are listed on page 229. Other books you may need are listed on page 230.

How to use this book

To diagnose a problem reported to you, start by reading Chapter 1. Then read other chapters and appendixes as appropriate.

Conventions

Throughout this book, the first (leftmost) byte or bit of a sequence is designated as byte or bit 0.

In programming syntax:

- [] indicates optional items.
- { } indicates selections or choices.

Summary of changes

Changes for GDDM Version 3 Release 2

This book includes the functions introduced with GDDM Version 3 Release 2.

Example output has been provided for PostScript and CGM traces from the TRCESTR external default; see “Example 9. Trace output containing PostScript output” on page 60 and “Example 10. Tracing an imported CGM” on page 62.

New messages have been added to Appendix C, “Message-to-module cross-reference” on page 185.

The RCP code for the DSFRCE call has been added (see Appendix E, “Request Control Parameter codes” on page 203).

The *GDDM Installation: Planning, Testing, and Servicing* books have been discontinued. The information in them has been moved into several other books.

Changes for GDDM Version 3 Release 1

This book includes the functions introduced with GDDM Version 3 Release 1.

The name of this book has been changed from *GDDM Diagnosis and Problem Determination Guide* to *GDDM Diagnosis*, and the chapter on debugging aids moved to the *GDDM Base Application Programming Guide*.

GDDM-PCLK and GDDM-OS/2 Link have become part of the Version 3 Release 1 base products for GDDM/MVS, GDDM/VM, and GDDM/VSE.

GDDM-REXX has become part of the Version 3 Release 1 base products for GDDM/MVS and GDDM/VM.

All references to GDDM/VMXA have been removed. With Version 3 Release 1, only one variety of GDDM (GDDM/VM) runs under VM/CMS.

GDDM-CSPF is not available with GDDM Version 3 Release 1.

An appendix listing the RCP codes used in GDDM Version 3 Release 1 has been added.

Note: Some of the information in this book was previously published in other GDDM books.

changes

Chapter 1. Diagnostic procedures

This chapter helps you determine whether a suspected problem is in one of the IBM GDDM licensed programs. If the problem is in GDDM, go to Chapter 8, “Reporting GDDM problems to IBM” on page 135, which tells you how to report the problem to the IBM Support Center staff.

For some problems, you may find that you do not have enough information to enable the IBM Support Center staff to help you. You may be asked to investigate the problem further by running a GDDM trace or by using the diagnosis facility of GDDM-IMD, GDDM-REXX GDDM-PCLK, or GDDM-OS/2 Link. Each of these aids is described in this book.

Initial diagnosis

If an unexpected result occurs while using GDDM, the cause of the problem could be:

- Device definitions
- An application program
- A GDDM utility program (for example, ICU, Vector Symbol Editor, Image Symbol Editor)
- The GDDM print utilities
- The output device or control unit
- The workstation customization
- The operating system, subsystem, or telecommunication access method
- The way GDDM has been installed or customized
- GDDM Base, GDDM-PGF, GDDM-IVU, GDDM-REXX, GDDM-GKS, GDDM-IMD, GDDM-PCLK, or GDDM-OS/2 Link.

Outlined below are the symptoms that are most likely to be met, together with some checks you can make that may find the cause.

Unexpected output

The output may be incorrect or missing altogether.

Incorrect output

If the output is not correct:

- If a *device token* has been used, it may be incorrectly defined or it may be the wrong device token for the device.
- There may be a problem with the *nickname* table. For example, the nickname may be directing the output to the wrong device.
- Check the application program being used. Is it device-dependent? If so, it may be running on the wrong device.
- Ensure that the devices being used are correctly defined to the system.

- Ensure that the DSOPEN statement has been correctly coded. If GDDM-GKS is being used, the equivalent is the open workstation (GOPWK) function call.
- Ensure that all the symbol sets required by the job are available.
- In GDDM-IVU, a user-defined map for a menu or help panel might contain an error. If your enterprise does not use standard maps, run a test using the standard maps. If the error is not reproduced, check the user-defined maps.
- For GDDM-GKS, check that the ADMMDFT GKSWs macro or statement gives the correct device token for the physical device.

No output

If the output is missing altogether, ensure that the:

- Output device (terminal, plotter, printer) is suitable for graphics or image output
- Device and control unit are at the correct level of microcode
- Device and control unit are correctly configured
- Devices are correctly connected, and the connections are not loose
- Devices are switched on and “online”
- Devices are correctly defined to the system
- Graphics are in segments, if the output device is a printer or plotter.

Device checks and sense codes

If these are received, the most likely causes are:

- The device is not defined, or is not defined correctly, to the subsystem or telecommunication access method.
- The device token or a nickname is causing the wrong data stream to be built.
- The device, the control unit, or both may be at an incorrect level of microcode.
- The device, the control unit, or both may be incorrectly configured.

Note: Under TSO, the device checks and sense codes may be normal, resulting from line-by-line reshow causing an incomplete data stream transmission before a complete reshow.

GDDM error messages

Error messages issued by GDDM Base, GDDM-IVU, GDDM-PGF, or GDDM-GKS begin with the prefix ADM. Other prefixes are:

AEM	GDDM-IMD
ERX	GDDM-REXX
GQD	GDDM-PCLK
GQF	GDDM-OS/2 Link

There are many causes for these messages. Check the message in the *GDDM Messages* book, and take any action recommended there.

Messages that start AFM are from GDDM-graPHIGS. Check the messages in the *Messages and Codes for graPHIGS* book.

Abends

If an abend is received, and diagnosis shows that the error is in GDDM, you might be asked to submit an APAR. For information about this, see Chapter 8, “Reporting GDDM problems to IBM” on page 135.

For a list of abend codes, see Appendix B, “Abend codes” on page 173.

Where to look for further advice

If you think you have isolated the problem, but have not fixed it, this section suggests possible sources of advice.

Problems with application programs

If you think that the problem may be caused by an error in the application program, but are not sure, the following may help you:

- The documentation supplied with the application program.
- The *GDDM Base Application Programming Guide*, which provides guidance information about the debugging task.
- Chapter 2, “GDDM tracing” on page 21 of this book, which describes the TRCESTR default keyword and FSTRCE call statements, which you can use to record a trace. Do this if you suspect an internal GDDM error.
- The *GDDM Base Application Programming Reference* and the *GDDM-PGF Programming Reference* books, which tell you the correct parameters and values to use with GDDM and GDDM-PGF call statements.
- The *GDDM Messages* book, which provides information about the messages issued by GDDM.

Problems with GDDM utility programs

The GDDM utility programs are:

- Image Symbol Editor (part of GDDM Base)
- GDDM-PCLK (part of GDDM Base)
- GDDM-OS/2 Link (part of GDDM Base)
- GDDM-REXX (part of GDDM Base in GDDM/MVS and GDDM/VM)
- Interactive Chart Utility (part of GDDM-PGF)
- Vector Symbol Editor (part of GDDM-PGF)
- Online Presentation System (part of GDDM-PGF)
- GDDM-IVU
- GDDM-IMD
- GDDM-GKS.

If there is a problem in one of these programs, use the online help information or tutorial, or look at the appropriate book to check whether the utility is being used correctly.

If any error messages appear, check them in the *GDDM Messages* book.

Problems with GDDM print utilities

If you suspect a problem with a GDDM print utility, inspect the system console log. Your system operator should have access to this, and may be able to tell from it the cause of the problem.

Errors will occur if the application program creates a print file that uses a symbol set that is not made available to the GDDM print utilities. This is a common cause of problems, so ensure that the symbol sets that have been specified are available to the print utility.

Problems with the hardware

The most common indication that the problem is hardware-related is that no graphics are displayed and the message ADM0275 GRAPHICS CANNOT BE SHOWN appears. The problem is usually caused by an incorrect device definition or an inappropriate controller configuration.

If you think there is a hardware problem, see the *GDDM System Customization and Administration* book. Ensure that the terminals being used can show graphics and check whether they require any additional features to do so. More information about hardware-related problems can be found in “What to do if things go wrong” on page 5.

If GDDM-IVU is being used, ensure that the device being used supports GDDM image functions. This information is listed for all devices in the *GDDM System Customization and Administration* book.

You may also need to refer to the component description or customization book for the devices involved.

Problems with customizing workstations

Failures that occur in a workstation may be caused by a problem with the customization of the workstation. If you suspect that the problem is of this type, see the *GDDM System Customization and Administration* book.

You should also check the customization procedures in the *Graphics Control Program User's Guide and Reference*, SC33-0207.

Note: This document only refers to Personal Computer/G(x) devices.

Problems with the operating system

If you think there is a problem with the operating system, refer to the procedures given in the installation book for your computer system or subsystem.

Check the minimum software levels required by GDDM; these are in the *General Information* manual for your computer system. You should also check any error messages in the appropriate books.

Problems with GDDM after its installation

The most likely symptoms of a problem arising after the installation of GDDM are:

- A program does not run.
- A program terminates abnormally.
- Several different programs fail.

Incorrect installation of GDDM may also mean that graphics are not shown on some, or all, of the devices that are capable of showing them.

If you suspect a problem with the installation of GDDM, look at the section “What to do if things go wrong.”

Problems within GDDM

The problem might be in GDDM, its associated programs, or an application program. If you believe the problem to be in GDDM or one of its associated licensed programs, report the error to IBM giving specific information. Chapter 8, “Reporting GDDM problems to IBM” on page 135 deals with this.

What to do if things go wrong

If you find that you cannot get GDDM to work satisfactorily, there are several things you should do before you report the problem to the IBM Support Center staff.

- If you are installing GDDM, check through the steps you have completed so far, and examine the console log for any unusual messages. Also review the entire installation process from the start, and check for errors. In particular check:
 - The preinstallation planning instructions
 - The VTAM bind parameters
 - The CICS table entries
 - The IMS table entries
 - Any defaults you have changed
 - The appropriate *GDDM Program Directory* and the preventive service planning (PSP) “bucket” for late information on the GDDM licensed programs you have installed.
- Check, in the section “Common installation errors and pitfalls” on page 12, for the symptoms of the problem. If you cannot find the problem described in that section, continue with these checks.

Ensure that you have the correct levels of:

- System
- Subsystem
- Access method
- Controller microcode
- Control program (where appropriate)
- Hardware features (specifically for graphics).

Check any relevant notes, restrictions, or prerequisites that are mentioned in the *GDDM Program Directory* for your computer system.

- If an error message appears, either on a display screen or on the console log, look it up in the *GDDM Messages* book.

- If an abnormal termination (abend) occurs, check Appendix B, “Abend codes” on page 173, which contains details of all GDDM abends.
- If some devices work successfully on GDDM, compare their definitions against the one you are now diagnosing.

If these checks show up no unusual circumstances, check your hardware; it could be that it is not set up correctly to show GDDM graphics. The section “Checking hardware characteristics” tells you how to do it.

If the problem remains, check with the IBM Support Center staff. For information about this, see Chapter 8, “Reporting GDDM problems to IBM” on page 135.

Checking hardware characteristics

To determine the cause of the problem, you may need to check the characteristics of the hardware at your enterprise. The following sections tell you how to do this, and may enable you to isolate the problem to a specific device or controller. Checking a screen takes about one minute, a controller another ten to fifteen minutes.

IBM 3179-G and 3192-G graphics diagnosis

First check that you have a 3179 Model G. No 3179 terminals, other than the 3179-G, can show graphics.

Next refer to the documentation provided with the terminal, *IBM 3179 Color Display Station Operator Reference and Problem Solving Guide*, GA18-2180.

Similar considerations apply to the 3192-G color display station.

IBM 3472-M graphics diagnosis

First check that you have a 3472 Model-M. No 3472 terminals, other than the 3472-M, show graphics.

Next refer to the documentation provided with the terminal, *IBM InfoWindow 3472: User's Guide*, GA18-2917.

IBM 3270 graphics diagnosis

This applies only to IBM 3278 and 3279 terminals.

Check that the terminal is built to display graphics.

To test for installed graphics features (programmed symbol sets 2 and 4):

1. Enter the Test mode by pressing Alt+TEST (the TEST key is on the bottom row of the key-pad at the left-hand end of the keyboard).
2. Release these keys, type /8, and press the ENTER key.

If graphics are available on the terminal, you can see the PS fields A through F with a symbol. Absence of a PS set is signified by a period. Triple-plane PS stores are marked with a red and white triangle.

3. To leave Test mode, press PF3.

Absence of the PS feature means that you need to have graphics capabilities installed on your terminal. Ask your system support personnel to verify the terminal

configuration. If you receive the stick man message “operation not permitted” in the Operator Information Area (OIA), check the controller for graphics (see the next section), and then review the screen functions again.

IBM 3270 EDCB verification

This section does not apply to IBM 3179-G color display stations or IBM 3270-PC workstations.

3270-devices that have been configured for extended functions – more correctly known as structured field and attribute processing (SFAP) – should normally operate with an extended device control block (DCB) allocated by the controller and created during customization.

As part of problem determination, perform the following test to confirm that the device is operating with an extended DCB. This test is more fully documented in the Error Codes Appendix of the *3274 Control Unit Description and Programmer's Guide*.

On any 3278 or 3279 display attached to the same controller as the display or printer being diagnosed:

1. Enter the Test mode by pressing Alt+TEST.
2. Release these keys, and type nn/6 to display the DCB for the device in question; nn is the coax port number in question (00 through 31). (If the device being used for the test is the port in question, /6 is enough.)
3. Press the ENTER key.

The first X'40' (64 decimal) bytes of the DCB in question are displayed on lines 3 through 6. The first two bytes of the displacement from the start of the control block of the data being displayed appear on line 2 (initially 00).

4. Continue to press the ENTER key.

Line 2 should change to 04, 08, 0C, 10, 14, 18, 1C for each pressing. Lines 3 through 6 change to display further bytes of the DCB.

If the test display stops at 1C, the device is operating with an extended DCB.

If the test display stops at 0C (with the keyboard inhibited with the minus function indicator on the fifth pressing of the ENTER key), the device is not operating with an extended DCB.

The reason for this lies with the controller, or the customization, or the features installed on the device. TEST /8 (displays), described above, or TEST 5 (printers), described below, can be used to verify the features installed on the device.

General printer diagnosis

If a problem occurs while using a printer, ensure that the printer has been set up with the correct page size. The printer page size is checked and changed using the printer operator panel; for more information, refer to the *Operating Instructions* for the printer. It is not important what lines per inch (lpi) or characters per inch (cpi) settings are used, as long as the resulting page size is correct.

Page depth in inches = maximum page length (MPL) / lpi

Page width in inches = maximum print position (MPP) / cpi

The other printer settings can be set as required.

If device tokens are being used, the page size defined to the printer must be at least as large as that defined to GDDM by the device token. (It does not matter if the printer has a larger page size defined to it than that defined by the device token.)

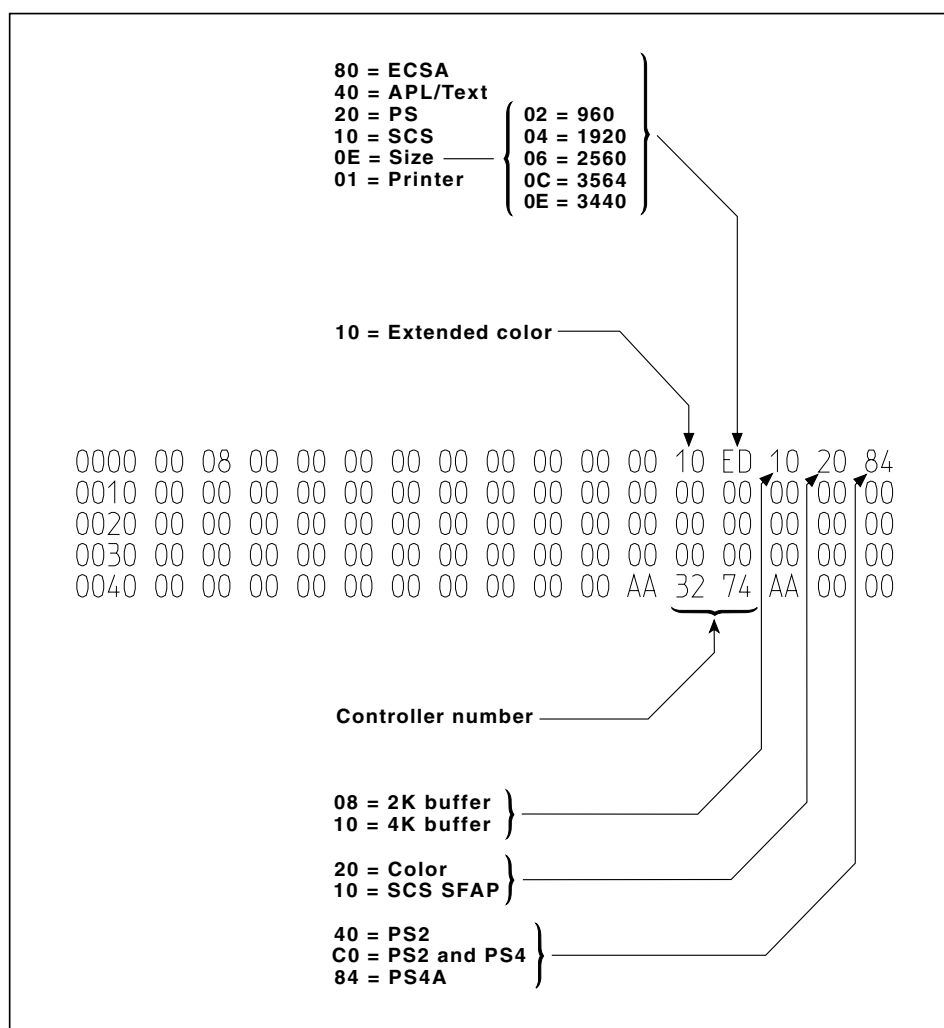
If device tokens for A4-sized output medium are being used on the IBM 4224 printer, the printer must be in full-page mode when the 4224 auto sheet feed (ASF) attachment is used. To check the mode, run test 307 and verify that 001 is displayed when Alt+9 are pressed.

IBM 3287 printer diagnosis

To test for installed graphics features on a 3287 printer, you must use the test procedure to print the print control information area (PCIA). To produce the PCIA:

1. Press and hold down the TEST button
2. Press and release the 5 button
3. Release the TEST button.

An example of the PCIA is shown below. For full details, see the maintenance information book that is shipped with the printer.



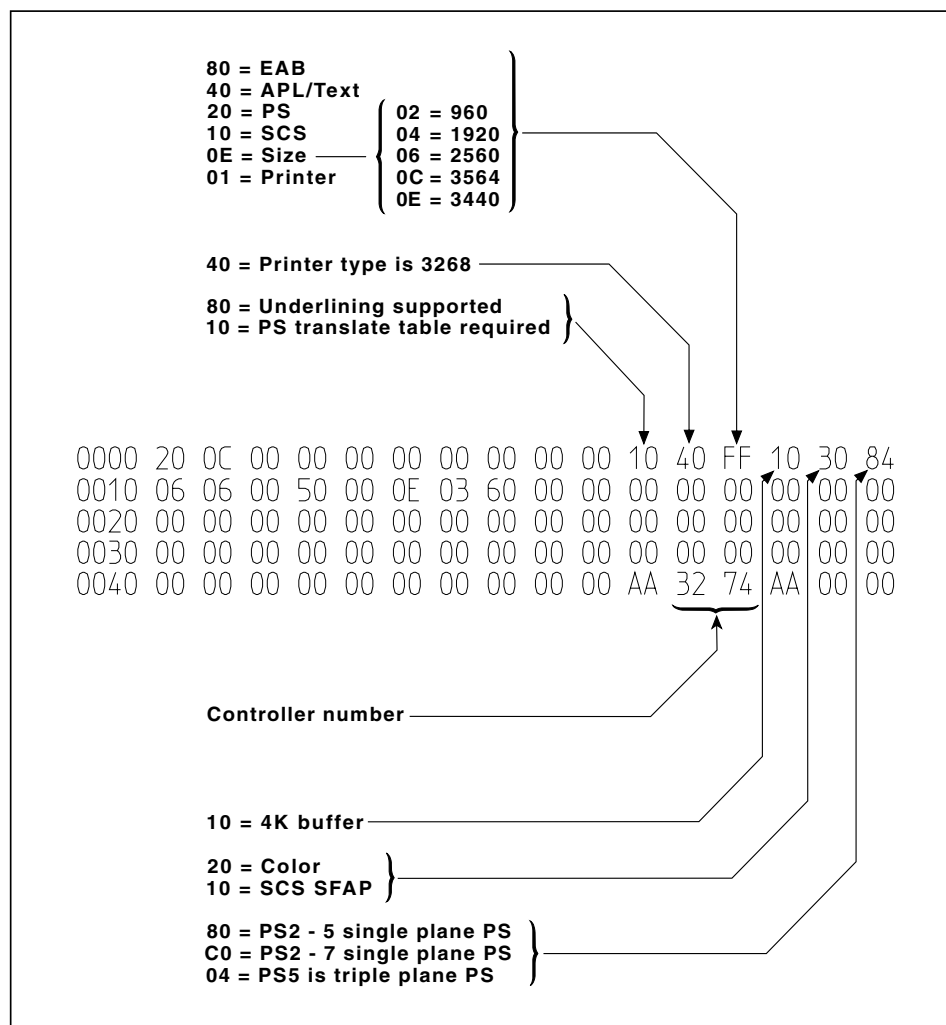
IBM 3268 printer diagnosis

To test for installed graphics features on a 3268 printer, you must use the test procedure to print the print control information area (PCIA). To produce this PCIA:

1. Press and hold down the TEST button
2. Press and release the 4 button
3. Release the TEST button.

An example of the PCIA is shown below. For full details, see the maintenance information book that is shipped with the printer.

Note: To enable use of the triple plane PS, language switch 1 must be in the On position.



IBM 3193 display station diagnosis

Check the stand-alone TEST procedure and the device setup procedure. The setup procedure enables you to change many attributes of the display; these may affect how the application runs. The setup procedures include:

- Whether extended attributes are supported, such as blink or reverse video.

- How many hardware partitions are available in each of the two logical terminals.
- The size of each logical terminal's viewport.
- The volume of the alarm.

Another potential problem is that Image is only supported on Logical Terminal 1 (LT-1).

IBM 3274 controller diagnosis

To permit graphics, a 3274 must be configured for graphics. Models eligible for graphics are:

- Any 3274 Model 31
- 3274-1A
- 3274-1C
- 3274-1D
- 3274-51C with enough storage

Models that are not eligible for graphics but that can be upgraded to be graphics-compatible are:

- Any 3274 Model 21
- 3274 Model 1B.

Ensure that you have the correct model and enough storage (96KB¹ are required for 3278, 3279, 3287, but later devices may need more storage; the IBM 3274 Model 31 has enough storage).

Then, check that the diskettes that have been customized are Configuration C or D for 3278, 3279, and 3287, or D or T for 3270-PC/G and /GX, 3290, and 5550. If they are not, get the level of support and customize as described in the *IBM 3274 Control Unit Planning, Setup, and Customization Guide*.

Specific questions, given in that guide, must be answered to support graphics. These can be any of the following. (The answers you must give are shown.)

- | | |
|-------|--|
| Q.121 | Is this correctly selected for your language requirement? |
| A. | 1 = yes |
| Q.161 | Color |
| A. | 1 = yes |
| Q.162 | SFAP |
| A. | 1 = yes |
| Q.163 | Extended Character Set Adapter (ECSA). |
| A. | 1 = yes for every device on the controller that has ECSA, but not greater than the number of category A terminals given in the answer to Q.112 |
| Q.164 | PS |
| A. | 1 = yes |

¹ 1KB equals 1024 bytes

- Q.165 Decompression.
 Normal recommendation is:
- A. 0 for 3274 A and D models,
 1 for 3274 C models.
- Q.166 Attribute Select Keyboard.
- A. C. This encompasses all options.
- Q.176 Decompression.
 Normal recommendation is:
- A. 1 This is for the BSC Enhanced Communication Option (Distributed Function Terminals) applied to 3270-PC/G and 3270-PC/GX. This is known as WACK support.

If you already have the 3274 correctly customized, use the modification procedure described in the *IBM 3274 Control Unit Planning, Setup, and Customization Guide* to verify that the above options are defined. Then complete the documentation card held with the controller for future reference.

IBM 3174 controller diagnosis

To enable graphics to be displayed on ASCII graphics terminals, an IBM 3174 controller with an asynchronous emulation adapter (AEA) is required. The 3174 licensed internal code must be at Configuration Support Release B2 or later.

The 3174 must be customized to suit the specific ASCII graphics terminal being used. This customization creates an AEA "Station Set" that defines the type and characteristics (for example, baud rate) of the terminal. Any local terminal-setup options must match the characteristics defined in the Station Set.

For terminals other than the DEC VT241 and Tektronix 4205, User Defined Tables (UDTs) are required in the 3174 AEA configuration. The graphics query reply field in the UDT must match one of the GDDM device token names for ASCII graphics terminals. These are:

DEC240	DEC330M	TEK4105	TEK4207	TEK4208M
DEC241	DEC340	TEK4205	TEK4207M	TEK4209
DEC330	DEC340M	TEK4205M	TEK4208	TEK4209M

UDTs are discussed in the *GDDM System Customization and Administration* book.

Notes:

- To use the full screen for GDDM graphics, set the alternate screen size in the 3174 AEA UDTs as follows:

DEC displays	0	(24 row only)
Tektronix 4105/4205	1	(30 row)
Tektronix 4207/4208/4209	2	(32 row)
- Set the graphics input line-out for Tektronix terminals to 400 milliseconds in the 3174 AEA UDTs.
- Tektronix graphics displays require XON/XOFF flow control to avoid corruption of graphics data. Configure this using question 731 in the 3174 AEA Station Set as well as in the display.

4. DEC VT330 and VT340 terminals should be set to VT300 mode with the status line set to HOST WRITABLE.

For details about defining the AEA Station Set (customization of the 3174, questions 721 and onward) and the UDT, see:

3174 Establishment Controller: Customizing Guide

3174 Establishment Controller: AEA Description and Reference

3174 Establishment Controller: Terminal User's Reference for Expanded Functions.

Non-IBM devices diagnosis

For non-IBM devices, refer to any applicable publications or consult your device supplier.

For ASCII graphics terminals, see "IBM 3174 controller diagnosis" on page 11.

Common installation errors and pitfalls

The following sections contain a list of common (and some not so common) errors and pitfalls that you may meet during the installation process. If you have a problem, check whether it is described below. Problems are listed in the following categories:

Problems associated with abends (on page "Problems associated with abends" on page 13):

The host software program is terminated with an abend code displayed or printed.

Problems associated with incorrect output (on page 13):

The output did not correspond to that expected, or did not appear.

Problems associated with messages (on page 14):

An unexpected message was issued.

Problems involving system performance (on page 18):

A reduction in performance occurred that could not be accounted for.

Problems involving device checks (on page 18):

A device "PROG" code or other code was displayed in the device's OIA.

Problems associated with SNA sense codes (on page 19):

An unexpected SNA sense code was encountered.

Problems associated with the Graphics Control Program (on page 20):

An unexpected Graphics Control Program problem was encountered.

Within each category, the errors and pitfalls are listed in order of the code, message number, or other characteristic associated with the error.

Check for the cause of the problem in any of the categories that may seem appropriate.

Problems associated with abends

For a list of abend codes, see Appendix B, “Abend codes” on page 173.

User abend code 1064

Symptoms:

User abend code 1064.

Possible causes:

This abend applies to GDDM features, for example GDDM-PGF or GDDM-IVU. It occurs when an attempt is made to use a feature that is:

- Not installed
- Not reinstalled when necessary
- Not installed correctly
- In an area of storage that cannot be accessed by the user.

Problem resolution:

If it is an installation problem, install or reinstall the feature correctly in an area that the user can access.

User abend code 1201, 2201, or G201

Problem resolution:

This abend occurs if an incomplete alphanumeric defaults module (ADMDATRN) is used with a 3179-G1, 3179-G2, or 3472-M color display station, or with a 3270-PC/G or /GX workstation or a 5550 multistation. See the section on compatibility with previous releases in the *GDDM Program Directory* for your computer system.

User abend codes 2053 and 2054

Symptoms:

User abend codes 2053 or 2054.

Applicable systems or subsystems:

IMS

Problem resolution:

These abends occur instead of messages ADM0001, ADM0002, or ADM0003. Register 15 at the time of the abend locates the corresponding message text. Look up the message in the *GDDM Messages* book.

Problems associated with incorrect output

Thick black lines on 3812 output

Symptoms:

Thick black lines appear on 3812 output.

Applicable systems or subsystems:

VM.

Problem resolution:

Either the required font has not been loaded, or the wrong level of the VM3812 program is in use.

User session logoff

Symptoms:

Missing Interrupt conditions and consequent user session logoff.

Applicable systems or subsystems:

VM.

Problem resolution:

For GDDM running under VM/ESA, the corresponding message is HCPMHT2150I, described in “Message HCPMHT2150I” on page 17.

Problems associated with messages

Messages beginning ADM

Symptoms:

ADM... Any message beginning with the letters ADM.

Problem description:

The message may appear on the display screen, on printer output, or in a console log.

Problem resolution:

Look up the message in the *GDDM Messages* book. The book contains an explanation of the message and an indication of what to do. Also, check below for additional information relating to the specific messages.

Message ADM0275, reason code 9

Symptoms:

ADM0275 GRAPHICS {(IMAGE)} CANNOT BE SHOWN, REASON CODE 9

Problem resolution:

Message ADM0275, reason code 9, is issued for a 3270-PC/G or /GX when the device has PS support but no graphics support.

Even when such a device has been customized with graphics support, it dynamically suppresses the graphics support when it is SNA-attached and its VTAM MODEENT SRCVPAC specification is incorrect. See the section on checking a VTAM network in the *GDDM System Customization and Administration* book.

Message ADM0275, reason code other than 9

Symptoms:

ADM0275 GRAPHICS {(IMAGE)} CANNOT BE SHOWN, REASON CODE n

Problem resolution:

Look up the message in the *GDDM Messages* book.

If the reason code shows that system tables are at fault, check the system tables, including VTAM bind definitions where appropriate.

If the reason code shows that the device is at fault, read the section “Checking hardware characteristics” on page 6.

If you are trying to display graphics using the GDDM-PCLK program, check the GDDM-PCLK procopt. Also make sure that you hot-key and do not press ENTER when the personal computer system is “opened” by the host computer application program. For more information, see the *GDDM Messages* book.

Messages beginning AEM

Symptoms:

AEM... Any message beginning with the letters AEM.

Applicable systems or subsystems:

All except IMS.

Problem description:

The message may appear on the display screen, on printer output, or in a console log. It comes from GDDM-IMD.

Problem resolution:

Look up the message in the *GDDM Messages* book. The book contains an explanation of the message and an indication of what to do.

Message DFS0089

Symptoms:

DFS0089I OUTPUT EXCEEDS BUFFER SIZE. LTERM NODE

Applicable systems or subsystems:

IMS.

Applicable environment:

Local, non-SNA attachment.

Problem resolution:

For local non-SNA displays under IMS, the OUTBUF parameter must be coded to define a buffer large enough to hold a complete output message. The maximum buffer size of 32KB is recommended for such displays. For more information, see the *GDDM System Customization and Administration* book.

Message DFS2078

Symptoms:

DFS2078 O/P REJECTED SENSE nnnnnnnn

Applicable systems or subsystems:

IMS.

Problem description:

Message displayed on IMS Master Console.

Problem resolution:

This message contains the sense/status bytes returned by VTAM. Check whether a device PROG error code has been displayed in the OIA. Then check the sections "Problems involving device checks" on page 18 and "Problems associated with SNA sense codes" on page 19 for information about the specific codes.

See the appropriate VTAM and hardware books for more information about the sense code.

Message DFS971I

Symptoms:

DFS971I I/O ERROR NODE, nnnnnnnn, xxxx

Applicable systems or subsystems:

IMS.

Problem description:

Message displayed on IMS Master Console.

Problem resolution:

This message relates to an I/O error communicating with a device. “nnnnnnnn” represents sense/status bytes returned by VTAM. For information about the specific code, see the section “Problems associated with SNA sense codes” on page 19.

For more information about the sense code, see the appropriate VTAM books.

Message DMKDID546I

Symptoms:

DMKDID546I INTERRUPTION <PENDING|CLEARED>

Applicable systems or subsystems:

VM.

Problem resolution:

Highly complex graphic output can sometimes incur a significant length of processing time in 3179-G1, 3179-G2, or 3472-M color display stations, or in 3270-PC/G or /GX workstations before they indicate the completion of an I/O operation. If the time involved exceeds the host computer system’s “missing interrupt time interval,” an error condition may be raised and the user session logged off. Under VM, the default interval is 30 seconds. For information about increasing this interval, see the *GDDM System Customization and Administration* book.

Messages beginning ERX

Symptoms:

ERX... Any message beginning with the letters ERX.

Applicable systems or subsystems:

Not VSE, not IMS, and not CICS.

Problem description:

The message can appear on the display screen, on printer output, or in a console log. It comes from GDDM-REXX.

Problem resolution:

Look up the message in the *GDDM Messages* book. The book contains an explanation of the message and an indication of what to do.

Messages beginning GQD

Symptoms:

GQD... Any message beginning with the letters GQD.

Problem description:

The message can appear on the display screen, on printer output, or in a console log. It comes from GDDM-PCLK.

Problem resolution:

Look up the message in the *GDDM Messages* book. The book contains an explanation of the message and an indication of what to do.

Messages beginning GQF

Symptoms:

GQF... Any message beginning with the letters GQF.

Problem description:

The message can appear on the display screen, on printer output, or in a console log. It comes from GDDM-OS/2 Link.

Problem resolution:

Look up the message in the *GDDM Messages* book. The book contains an explanation of the message and an indication of what to do.

Message IST211I

Symptoms:

IST211I NCP SLOWDOWN INITIATED FOR

Applicable systems or subsystems:

CICS under MVS, or IMS.
VM and VSE.

Applicable access methods:

VTAM remote.

Problem resolution:

In a remote SNA configuration, graphic data streams can affect response times for non-GDDM users, and can initiate NCP SLOWDOWN conditions. You should review your NCP generation, as described in the *GDDM System Customization and Administration* book, paying particular attention to buffer, PACING, and VPACING parameters. If in doubt, specify PACING=(2,1) and VPACING=2 on relevant SNA PU macros in your NCP generation.

Message HCPMHT2150I

Symptoms:

HCPMHT2150I devtype addr AN INTERRUPT IS PENDING

Applicable systems or subsystems:

VM.

Problem resolution:

Highly complex graphic output can sometimes incur a significant length of processing time in 3179-G1, 3179-G2, or 3472-M color display stations, or in 3270-PC/G or /GX workstations before they indicate the completion of an I/O operation. If the time involved exceeds the host system's "missing interrupt time interval," an error condition may be raised and the user session logged off. Under VM, the default interval is 30 seconds. For information about increasing this interval, see the *GDDM System Customization and Administration* book.

Problems involving system performance

Line time-outs

Symptoms:

Line time-outs.

Applicable access methods:

Remote (link-attached) non-SNA environments.

Problem resolution:

Telecommunication line time-outs may occur for a 3179-G1, 3179-G2, or 3472-M color display station, or for a 3270-PC/G or /GX workstation, or a 5550 multistation if the device is BSC-attached but the associated 3274 controller has not been configured with WACK support. For more information, see the *GDDM System Customization and Administration* book.

Missing interrupt conditions

Symptoms:

Missing interrupt conditions and consequent user session logoff.

Applicable systems or subsystems:

VM.

Problem resolution:

For GDDM running under VM/ESA, see the description in “Message HCPMHT2150I” on page 17.

NCP SLOWDOWN conditions for non-GDDM users

Symptoms:

Poor response times for non-GDDM users, with NCP SLOWDOWN conditions initiated.

Applicable systems or subsystems:

CICS under MVS, or IMS.

Applicable access methods:

VTAM remote.

Problem resolution:

See the description in “Message IST211I” on page 17.

Problems involving device checks

Machine check 207

Symptoms:

Machine check 207.

Problem resolution:

This machine check may be indicated on a 3270-PC/G or /GX if required patches or customization options have not been applied to the associated 3274 controller. See the *GDDM System Customization and Administration* book.

Device check PROG752

Symptoms:

PROG752.

Problem resolution:

This device PROG error code is usually caused by an incorrect device definition. Ensure that the terminal definition in the GDDM System Definition database matches the device characteristics (as configured, where appropriate). Under IMS, also ensure that:

- The terminal definition and device characteristics match the IMS TERMINAL macro SIZE operand and, where applicable, the VTAM bind definition (in particular, the screen size)
- The VTAM bind definition conforms to those shown in the *GDDM System Customization and Administration* book.

Program check

Symptoms:

Program checks in mixed fields.

Problem resolution:

Changes in data stream in fields that include a mixture of single-byte and double-byte character sets (SBCS/DBCS) may result in program checks on 5550 systems. If the subsystem and communications functions permit, this problem can be avoided by using a sufficiently large transmission buffer.

Problems associated with SNA sense codes**Sense code 1005**

Symptoms:

Sense code 1005.

Applicable systems or subsystems:

IMS, or TSO, or CICS under MVS.

VTAM under VM or VSE.

Problem resolution:

This SNA sense code is usually caused by an incorrect device definition. Ensure that the terminal definition in the GDDM System Definition database matches the device characteristics (as configured, where appropriate). Under IMS, also ensure that:

- The terminal definition and device characteristics match the IMS TERMINAL macro SIZE operand and, where applicable, the VTAM bind definition (in particular, the screen size)
- The VTAM bind definition conforms to those shown in the *GDDM System Customization and Administration* book.

Sense code 800A

Symptoms:

Sense code 800A.

Applicable systems or subsystems:

IMS, or TSO, or CICS under MVS.

VTAM under VM or VSE.

Applicable access methods:

VTAM in a remote non-SNA environment.

Problem resolution:

This SNA sense code can arise if you try to use GDDM on a non-SNA remote VTAM-attached terminal under IMS.

GDDM is supported under IMS through VTAM, but only for local or SNA-attached terminals.

Problems associated with Graphics Control Program errors

User symptoms

The Graphics Control Program “hangs” with X clock in the OIA during a nonretained host graphics outgoing data stream. SY005 component informational 08690000 occurs unnecessarily on a /GX during a nonretained host graphics outgoing data stream. The drawing performance of a nonretained host graphics outgoing data stream may also be adversely affected by this problem.

Problem solution

The problem occurs only if the segment storage exceeds 86KB, and it can be bypassed by recustomizing with a different segment store value. However, it could then still occur with a different data stream if the segment store remains above 86KB. Because segment storage is limited to 63KB, this problem cannot occur on a 3270-PC/GX with Graphics Control Program 1.12 or Graphics Control Program 2.10.

Chapter 2. GDDM tracing

Product-sensitive programming interface

You can choose to run a GDDM trace as part of your diagnosis procedure, or the IBM Support Center staff may ask you to run a trace, to provide more information to help solve the problem. In the latter instance, you may be provided with trace statements to include in the application program.

From Version 2 onward, GDDM provides an improved system of tracing. The methods used previously are still valid, and are described later in this chapter. However, the recommended way to run a GDDM trace is to write a set of statements to be included in the external defaults file, using the GDDM default keyword TRCESTR, like this:

```
[label] ADMMDFT TRCESTR='IF TSI THEN DO           '
        ADMMDFT TRCESTR='  FLOW ;                 '
        ADMMDFT TRCESTR='  FULLIO ;               '
        ADMMDFT TRCESTR='  TIME                   '
        ADMMDFT TRCESTR='END                       '

```

This chapter describes what you can do with GDDM TRACE, how to invoke it, and how to write TRACE statements. It describes the functions you can specify and the format of trace output. Examples of TRACE statements, together with the output produced by them, are provided. The chapter then describes the methods of tracing that were previously available and that can still be used. Finally, there is a section that tells you how to locate GDDM control blocks, including the in-storage trace table, from a dump.

Notes:

1. A GDDM trace can produce large amounts of output. Try to be as restrictive as possible when specifying the part of the program to be traced and the amount of output needed.
2. For help in analyzing the data produced by the trace, see also *3270 Information Display System: Data Stream Programmer's Reference* and *Intelligent Printer Data Stream Reference*.
3. Although the module names for the GDDM print queue manager (on MVS/TSO) have the prefix ADM, its modules cannot be traced with the GDDM trace methods because it is an ISPF application. Use the TSO TEST or ISPF DIALOG TEST facilities, which are described in:
 - *TSO Command Reference*, SC28-1881
 - *ISPF User's Guide*, SC34-4484

The TRCESTR external default keyword

When tracing with TRACE, you can specify:

- The level at which the trace is to run (module, component, subcomponent, API)
- The component or module to be traced
- Whether to trace on entry to, or exit from, the item being traced, or both
- The number of times a particular set of conditions must occur before a trace record is produced
- Particular occurrences or ranges of events to be traced
- Your own abend codes
- The type of trace records to be produced
- Whether the contents of floating-point, of general-purpose registers, or of both are to be included in the trace output
- Whether to include storage use in your trace output
- A full or partial I/O trace
- 5080 tracing.

Starting trace under different subsystems

This section tells you how to start tracing in each of these system environments:

- TSO and the TSO print utility
- CMS
- CICS
- IMS
- VSE/Batch.

It also covers tracing in an application program.

Tracing under TSO

The most flexible way to start tracing is with an external default file. To use this method:

1. Allocate the ddname ADMTRACE to SYSOUT, or to a data set large enough to receive the trace output. It is not necessary to supply DCB characteristics for this data set because GDDM reformats it.

If the output is going to a tape, the data set must be allocated with DSORG=PS, LRECL=132.

2. Place your TRCESTR statements in an external defaults data set. The ddname for the external defaults data set is ADMDEFS. The DCB characteristics must be LRECL=80, RECFM=FB.
3. Allocate the ddname ADMDEFS to your defaults data set.
4. Run the application program or utility that you want to trace.

Remember to disable tracing when you no longer require it. You can do this by erasing the TRCESTR statements from your defaults file, or by turning them into

comments. To comment them out, place an asterisk (*) in column 1 of each TRCESTR default statement.

Tracing in the TSO print utility environment

When tracing in the TSO print utility environment, trace records are written using sequential files. The trace output for the utility subtask (ADMOPST) associated with any one printer is written using a sequential file with a ddname the same as the LUNAME of the printer. These ddnames should be assigned to suitable data sets or SYSOUT destinations in the JCL for the utility. In this way, the trace output from the utility can be limited to that associated with a specific printer.

Tracing under VM

The most flexible way to start tracing is with an external default file. To use this method:

1. Place your trace statements in your defaults file, PROFILE ADMDEFS, and ensure that the file is on your A-disk.
2. Specify a destination for the trace output.

The trace records are normally written to a sequential file with the file-identifier ADM00001 ADMTRACE A1. If you want to change the file-name or file-type used, you can do so by specifying the CMSTRCE external default in the source format defaults file. If the CMSTRCE value is set to blank, the trace records are written to the virtual printer, using spool file processing. To direct trace output to the virtual printer, you should include the following default specification in the external defaults file:

```
[label] ADMMDFT CMSTRCE=(,)
```

If you do not have enough space on your A-disk, you should first spool the printer HOLD using the CP SPOOL command, and then spool your trace output to a virtual printer.

3. Run the application program or utility that you want to trace.
4. When tracing is complete, spool the virtual printer NOHOLD if you now want to print the trace output.

Remember to disable tracing when you have finished. You can do this by erasing the TRCESTR statements from your defaults file, or by commenting them out. To comment them out, place an asterisk (*) in column 1 of each TRCESTR external default statement.

Tracing under CICS

Under CICS, it is generally not possible to create a permanent external defaults file, like that available to TSO or VM users. However, a restricted external defaults file mechanism is available to CICS users, mainly for debugging purposes.

To run GDDM trace under CICS:

1. Determine the CICS identification of the terminal on which you intend to run GDDM. You can get this terminal ID using the CEMT transaction.
2. Allocate a data set to receive the trace output. Under CICS/ESA systems, if the output is going to a tape, the data set must be allocated with DSORG=PS.

The trace records are written by standard CICS services to a single transient data destination, with a default name of ADMT. This must be defined in the CICS destination control table (DCT) in a way that suits the requirements of the users at your enterprise. Typically, the destination would be defined as an extrapartition destination, which would direct the trace records to an external data set for printing later. This destination name can be changed by altering the CICTRCE external default.

3. Create a temporary storage queue, containing the replacement defaults. This queue must have a name of the form ADMDxxxx, where xxxx is the CICS terminal ID obtained by running the CEMT transaction in step 1 on page 23. (The prefix ADMD can be changed in a defaults module or through a SPINIT call statement, using the external default CICDFPX=aaaa.)

If you are authorized to do so, you can create a temporary storage queue using the CECI transaction. This transaction is documented in the *CICS Application Programming Guide*, which includes a discussion of the security and authorization aspects of the transaction.

The replacement defaults must be in source format, as described in the *GDDM System Customization and Administration* book.

For example, to turn trace on, enter these transactions:

```
CECI DELETEQ TS QUEUE('ADMDxxxx')
CECI WRITEQ  TS QUEUE('ADMDxxxx')
               FROM(' ADMMDFT TRCESTR="FLOW"')
```

Notes:

- a. xxxx is the CICS terminal ID from step 1 on page 23.
- b. You must have a blank before ADMMDFT and you must include any embedded strings with the TRCESTR parameter in double quotation marks or a pair of single quotation marks, and it must be in uppercase.

If you have more than one TRCESTR statement, you must code a FROM statement for each ADMMDFT statement, thus:

```
CECI DELETEQ TS QUEUE('ADMDxxxx')
CECI WRITEQ  TS QUEUE('ADMDxxxx')
               FROM(' ADMMDFT TRCESTR="IF TSI THEN FLOW"')
               FROM(' ADMMDFT TRCESTR="PARTIO"')
```

4. Run the application program or utility that you want to trace.

Remember to disable tracing when you have finished. The external defaults you have established remain in effect until the temporary storage queue is deleted, either explicitly, or automatically at CICS termination. They are not normally retained after CICS termination.

Tracing under IMS

Under IMS, you cannot use an external source defaults file. Therefore, you must include your trace statements in the external defaults module (ADMADFI). This module is either link-edited with your application program or GDDM utility program, or is in your library data set. How to change ADMADFI is described in the *GDDM System Customization and Administration* book.

Run a GDDM trace as follows:

1. Allocate a data set to receive the trace output. If the output is to go to a tape, the data set must be allocated with DSORG=PS.

The trace records are written using a sequential file with a default name of ADMTRACE. This should be allocated to a suitable data set or SYSOUT destination before GDDM is invoked.

2. Ensure that your trace statements are included in the external defaults module.
3. Run the application program or utility that you want to trace.

Remember to disable trace, when you no longer want to run it, by restoring the defaults module to its original state.

Tracing under VSE/Batch

The most flexible way to start tracing is to include tracing defaults in the job stream, immediately after the invocation of GDDM. To use this method:

1. Ensure that you have allocated a suitable VSAM data set to receive the trace output.
2. Include this statement in your job stream:

```
// DLBL ADMTRACE,'your_output_file_name',,VSAM
```
3. Include any TRCESTR statements (and other external defaults or nicknames) immediately after the invocation of GDDM.

Remember to disable tracing when you no longer require it. You can do this by erasing the TRCESTR statements from the job stream, or by commenting them out. To comment them out, place an asterisk (*) in column 1 of each TRCESTR default statement.

Tracing in an application program

Tracing can be initiated by an application call, for example, ESEUDS. Details of this method are given in “Other tracing methods” on page 69.

Note: The trace statements required for tracing from an application call differ from those required when using an external defaults file or module:

- The trace statements in an external file or module can be spread over several lines or ADMMDFT statements.
- A single trace statement in an application call must conform to the syntax for a complete program. For example, all parts of an IF THEN ELSE statement must be within a single application call. However, a trace specification can be built up from any number of separate application calls. These calls are appended to the trace program formed from an external file, if there is one.

Tracing multiple instances of GDDM

If you are running under TSO or VM, the GDDM TRCESHR default facility enables you to trace more than one instance of GDDM to the same trace file. This is particularly useful in a task-manager environment, where you may have more than one application to be traced at the same time. You may also want to trace combinations of task managers and applications.

- If only one instance of GDDM is to be traced, tracing proceeds as usual.
- If two or more GDDM instances are to be traced and each instance uses a different trace file, there is no contention for the trace files and tracing proceeds as usual. Each instance can be directed to use a different trace file with suitable trace output defaults.

For example, in one instance of GDDM use:

```
CMSTRCE=(INST0001,ADMTRACE)
```

and in the second instance use:

```
CMSTRCE=(INST0002,ADMTRACE)
```

- If two GDDM instances are traced, and both instances try to use the same trace file, the second GDDM instance to attempt tracing is unsuccessful, because the trace file can only be shared serially.
- You can code a GDDM TRCESHR default statement to specify that a trace file is to be shared among instances of GDDM. If an instance of GDDM is initialized with this default set, both it and subsequently initialized instances of GDDM that specify tracing to the same trace file use the one file.

The format of the TRCESHR statement is:

```
[label] ADMMDFT TRCESHR=YES|NO
```

Use ADMMDFT TRCESHR=YES to specify that a trace file is to be shared.

There must be a blank before ADMMDFT.

Note: Specifying this default means that the instance of GDDM is to be a coordinator for shared use of the trace file. The coordinating instance must only be terminated after all other instances sharing the trace file have been terminated.

Trace file records are prefixed with a three-digit number that identifies the instance of GDDM that generated the record. The numbers are assigned to the instances of GDDM in the order in which the instances begin to use the trace file. An example of such trace output is shown on page 57.

Sample trace statements

A number of sample trace statements are shipped with GDDM on the installation tape. The statements are provided as comments. In other words, they are preceded by an asterisk (*). If you need to take a trace of a GDDM application, copy the samples into another file that you have created yourself, remove the * from the required statements, and run the application again.

The trace statements can be run as they are. However, if the IBM System Support Center staff ask you to run a GDDM trace, they may also ask you to edit one or more of the sample statements, and then rerun the application to provide them with more information.

- Under MVS, the sample trace statements are held in a member called ADMUTRCE in the SADMSAM partitioned data set.
- Under VM, they are held in a file called ADMUTRCE ADMDEFS.
- Under VSE, they are held in the Z sub-library.

Trace keywords

The table below shows the source syntax and defaults for each of the keywords associated with tracing.

Keyword	Meaning of the keyword	Source syntax of the keyword	GDDM default
TRCESTR	Trace options	TRCESTR='xxxxxxxxx'	None
TRCEWID	Trace output width	TRCEWID={SINGLE DOUBLE}	SINGLE
TRTABLE	Trace table size, in-core	TRTABLE=<i>n</i>	100
TRCESHR	Trace share	TRCESHR={NO YES}	NO
TRACE	Trace word value	TRACE={0 <i>n</i>}	0
CICTRCE	Trace output transient data name	CICTRCE=aaaa	ADMT
IMSTRCE	Trace output ddname	IMSTRCE=aaaaaaaa	ADMTRACE
TSOTRCE	Trace output ddname	TSOTRCE=aaaaaaaa	ADMTRACE
CMSTRCE	Trace output (file name,file type)	CMSTRCE=(aaaaaaaa,bbbbbbbb)	(ADM00001, ADMTRACE)

Details of coding TRCESTR statements begin below. Full details of the syntax of TRCESTR statements are in Appendix D, “Trace-string syntax” on page 199.

Details of the TRCEWID statement are on page 42.

The other keywords are described in the *GDDM System Customization and Administration* book.

Coding TRCESTR statements

A set of TRCESTR statements consists of one or more

```
ADMMDFT TRCESTR='xxxxxxxxxxxxx'
```

statements in an external default module or file. Note that:

- You must have a space before ADMMDFT.
- If you are including your TRCESTR statements in an external default *file*, rather than a default *module*, you can begin the statements with DEFAULT instead of ADMMDFT.
- Each character string can be up to 256 characters long.
- All the TRCESTR statements in the defaults module or file are used to make up a single trace program; this program must conform to the syntax defined in Appendix D.

A set of trace statements can consist of one action or several actions to be executed whenever the trace processor is invoked. It can be as simple as:

```
[label] ADMMDFT TRCESTR='FLOW'
```

This statement specifies that trace is to be invoked and that standard trace statements (known as flow statements) are to be produced in the trace output. Every module boundary is traced on entry and exit.

Remember that the ADMMDFT statement must *not* begin in column 1.

You can restrict the amount of trace output produced. If, for example, you want trace output from the ICU only, you can specify:

```
[label] ADMMDFT TRCESTR='IF ICU THEN FLOW'
```

You can specify several actions in one statement. For example, the statement:

```
[label] ADMMDFT TRCESTR='IF ICU THEN FLOW PARTIO TIME'
```

requests a partial transmission trace from the ICU, with standard output. Specifying TIME causes a time stamp to be included in the trace output for each traced module.

The general form of a statement is:

```
[label] ADMMDFT TRCESTR='IF event THEN actions ELSE actions'
```

The event and action functions that you can specify are defined in “Functions available with the TRCESTR keyword” on page 30.

Compound statements are allowed. Use semicolons to separate all statements within the compound statement. The previous example could be written:

```
[label] ADMMDFT TRCESTR='IF ICU THEN DO FLOW; PARTIO; TIME END'
```

IF statements can be nested:

```
[label] ADMMDFT TRCESTR='IF ESI THEN IF COUNT(4) THEN DO FLOW;'  
ADMMDFT TRCESTR='PARTIO; TIME END'
```

They can be nested within compound statements:

```
[label] ADMMDFT TRCESTR='IF TSI THEN  
ADMMDFT TRCESTR=' DO  
ADMMDFT TRCESTR=' FLOW ;  
ADMMDFT TRCESTR=' IF NAME(''ADMLN*'' ) THEN  
ADMMDFT TRCESTR=' DO  
ADMMDFT TRCESTR=' LIST(100,200);  
ADMMDFT TRCESTR=' PARTIO(ON)  
ADMMDFT TRCESTR=' END;  
ADMMDFT TRCESTR=' TIME  
ADMMDFT TRCESTR=' END'
```

Note that, if you specify a variable character string in a TRCESTR statement, such as ADMLN* in the statement using the NAME event function above, you must include the character string in double quotation marks or a pair of single quotation marks, and it must be in uppercase.

The relational expression in an IF statement can consist of a combination of events. For example:

```
[label] ADMMDFT TRCESTR='IF CMPNT & (ICU | FSM) THEN FLOW'
```

This example can also be written:

```
[label] ADMMDFT TRCESTR='IF CMPNT AND (ICU OR FSM) THEN FLOW'
```

Signed numeric values can be expressed as binary numbers (B'011100'), decimal numbers (-123445) or hexadecimal numbers (X'789AB').

The contents of an address (%), a general-purpose register (GR), or a floating-point register (FR), can be referred to in a relational expression or as a parameter:

```
[label] ADMMDFT TRCESTR='IF ENTRY & ESI THEN LIST((1 GR +4)%,4)'
ADMMDFT TRCESTR='IF 13 GR >= X''0F1B2'' THEN FLOW'
```

Note that, if you use % to access an address for which trace does not have read access, an abnormal termination occurs in module ADMATPX for the unacceptable address.

You can trace on a component RCP code. These are listed in Appendix E, “Request Control Parameter codes” on page 203. For example, to trace entry and exit parameters for ASREAD:

```
[label] ADMMDFTRCCESTR='DO IF CMPNT THEN
ADMMDFTRCCESTR=' IF (1 GR +4)%%=X'C100000' THEN
ADMMDFTRCCESTR=' PARMSF END
```

In this example, nesting ensures that the address is evaluated *only after* CMPNT is satisfied, thus eliminating the risk of an abend.

All the relational expressions in an IF statement are evaluated on every occasion (no jumping code); thus the following two examples are equivalent:

```
[label] ADMMDFTRCESTR='IF COUNT(4) & LIMIT(2,12) THEN FLOW'
ADMMDFTRCESTR='IF LIMIT(2,12) & COUNT(4) THEN FLOW'
```

If you want the effect that would be produced by jumping code, you must explicitly code nested IF statements:

```
[label] ADMMDFTRCESTR='IF COUNT(4) THEN IF LIMIT(2,12) THEN FLOW'
ADMMDFTRCESTR='IF LIMIT(2,12) THEN IF COUNT(4) THEN FLOW'
```

You can use the following operators in TRCESTR statements – where words are used, uppercase and lowercase letters are both acceptable:

+	add	=	equal to
-	subtract	≠	not equal to
*	multiply	<	less than
/	divide	>	greater than
&	and	≤	less than or equal to
	or	≥	greater than or equal to
¬	not		

The % sign can be used to indicate the contents of an address, as described on page 29. It is also used as a token on the PRINT function, as described on page 35.

Note: If you are compiling or assembling the TRCESTR statements, ampersands (&) within the statements must be repeated. For example:

```
[label] ADMMDFT TRCESTR='IF CMPNT && ICU THEN FLOW'
```

You can put a **CLEAR** statement at the start of your set of **TRCESTR** statements, and a **FORCE** statement at the end. These control the priority if you have more than one set of **TRCESTR** statements. The next section of this chapter gives examples of these functions.

Functions available with the TRCESTR keyword

In TRCESTR statements, you can use:

- A CLEAR function
- A FORCE function
- Action functions
- Event functions
- Control functions.

These are described in the following sections. Working examples that show sample trace output are provided on page 44.

CLEAR

You can start your set of TRCESTR statements with a CLEAR statement:

```
[label] ADMMDFD TRCESTR='CLEAR          '
          ADMMDFD TRCESTR='IF ENTRY THEN FLOW  '
```

If you specify CLEAR, any preceding TRCESTR statements are ignored, unless FORCE has already been invoked. You would normally use CLEAR only in an application call for tracing, for example, an ESSUDS call.

FORCE

At the end of your set of statements, you can specify FORCE:

```
[label] ADMMDFD TRCESTR='IF ENTRY THEN FLOW  '
          ADMMDFD TRCESTR='FORCE              '
```

This causes subsequent TRCESTR statements and subsequent calls to FSTRCE to be ignored, even if they contain the CLEAR keyword. That is, FORCE overrides all application FSTRCE calls and non-defaults-file tracing.

Action functions

The action functions that you can specify are summarized in this table:

Function	Arguments
ABEND	(<i>number</i>)
CDPDS	[({ ON OFF <i>control</i> })]
CGMREP	[({ ON OFF <i>control</i> })]
DSOPEN	[({ ON OFF <i>control</i> })]
FLOW	none
FREGS	none
FULLIO	[({ ON OFF <i>control</i> })]
FULLTCA	[({ ON OFF <i>control</i> })]
GREGS	none
HRIG	[({ ON OFF <i>control</i> })]
LIST	(<i>low_address</i> [, <i>length</i>])
NULL	none
PARMSF	(<i>number</i>)
PARTIO	[({ ON OFF <i>control</i> })]
POSTSCR	[({ ON OFF <i>control</i> })]
PRINT	("control_string" [, <i>remaining_parameters</i>])
STGREP	[({ ON OFF <i>control</i> })]

Function	Arguments
TIME	none
TRACE	(<i>number</i> , ON OFF)
5080IO	[({ ON OFF <i>control</i> }[, <i>number</i>])]

All the actions for which you can specify a control parameter are *latched*, that is, the action is inactive until you specify it to be ON. It then remains ON until you specify it to be OFF. Details of the control parameter are given on page 42.

The order in which the actions take effect (that is, place information in the trace output file) is fixed to minimize duplication. For example, if two separate conditions are satisfied, both of which call for a time stamp, only one time stamp is placed in the trace file. The order in which the actions are evaluated is:

PRINT LIST	These actions take effect in the order in which they are evaluated by the trace interpreter.
FLOW PARMSF TIME TRACE GREGS FREGS STGREP ABEND	
CDPDS CGMREP DSOPEN PARTIO FULLIO FULLTCA HRIG POSTCR 5080IO	The effect of these latching actions is not directly related to their evaluation by the trace interpreter.

A detailed description of each action function is given below, in alphabetic order.

ABEND

ABEND(*number*)

GDDM is terminated abnormally with the specified abend number. The *number* must be a decimal number in the range 1 through 999.

CDPDS

CDPDS[({**ON**|OFF|*control*})]

If ON, this specifies that the composite-document presentation data stream (CDPDS) input is to be included in the trace output. The structure and content of

the CDPDS are described in the *GDDM Base Application Programming Reference* book.

If the header of a CDPDS structured field is invalid, the output includes a note of this and the first eight bytes of the field. If a field is less than eight bytes long, the output includes a note of this and the short record.

The *control* function is described on page 42.

An example of trace output containing CDPDS input is shown on page 58.

CGMREP

CGMREP[({ONOFF|control})]

If set to ON, this specifies that information relating to the CGLOAD and CGSAVE conversions is to be included in the trace output. This information includes:

- The conversion profile used
- The CGM and GDF orders found or created
- Changes to CGM orders generated, for CALS compliance

The *control* function is described on page 42.

DSOPEN

DSOPEN[({ONOFF|control})]

If ON, nickname processing within a DSOPEN call is traced so that it can be checked.

The *control* function is described on page 42.

FLOW

FLOW

This specifies that standard trace records are to be included in the trace output. Standard trace records are those produced at all module entry and exit points.

Note: If the FLOW option is active, SLIP IF traps set to the start of a GDDM module are not effective. This is because the trace modules alter the entry point into the GDDM module, changing the normal save area chaining so as to catch the exit from a module in the trace.

FREGS

FREGS

This specifies that the contents of the four floating-point registers (FREGS) are to be included in the trace output.

FULLIO

FULLIO[({ONOFF|*control*})]

If ON, this specifies that a full transmission trace is required. The full I/O buffer is included in the trace output for each transmit operation and each receive operation.

The *control* function is described on page 42.

FULLTCA

FULLTCA[({ONOFF|*control*})]

If ON, this specifies that a full transmission trace with terminal services interface control area (TCA) blocks is required.

The key blocks in the TCA are the terminal descriptor block (TDB) and the terminal request block (TRB); these are described on pages 166 and 168, respectively.

The *control* function is described on page 42.

GREGS

GREGS

This specifies that the contents of the sixteen general-purpose registers (GREGS) are to be included in the trace output.

HRIG

HRIG[({ONOFF|*control*})]

If ON, this specifies that the output data stream directed to a page printer is to be included in the trace output. An example of trace output containing this data is shown on page 59.

The *control* function is described on page 42.

LIST

```
LIST(low_address[, length])
```

This specifies an area of storage to be listed in dump format.

The *low_address* parameter shows the target location of the beginning of the required storage area, and the optional *length* parameter specifies the length, in bytes, of the storage area to be listed. If the *length* is less than 1 or is not specified, 4 bytes are listed.

Note: If you try to display storage to which trace does not have read access, an abnormal termination occurs in module ADMATPX for the unacceptable address.

NULL

```
NULL
```

The NULL function performs no action. It is provided for use in nested IF THEN ELSE statements.

PARMSF

```
PARMSF[(number)]
```

This specifies that parameters of any external (API) calls are to be included in the trace output, in formatted notation. For any component calls other than API calls, or if the *number* parameter is set to zero, PARMSF produces the request control parameter (RCP) code and function title only. Subcomponent and module level tracing does not produce any output from PARMSF.

The optional *number* parameter specifies the maximum number of bytes to be listed for each traced parameter. If it is not specified, a default limit of 80 bytes is applied. If *number* is negative, zero is assumed.

PARTIO

```
PARTIO[({ON|OFF|control})]
```

If ON, this specifies that a partial transmission trace is required. The data in the I/O buffer, up to a maximum of 16 bytes, is included in the trace output for each transmit operation and each receive operation. If the I/O buffer data exceeds 16 bytes, the first 16 bytes are included in the trace output with an indication that the data has been truncated and the length of the full data.

The *control* function is described on page 42.

POSTSCR

```
POSTSCR[({ON|OFF|control})]
```

If ON, this specifies that PostScript debug information be included in the trace output. To include the PostScript output in the trace (in EBCDIC format), specify the FULLIO function.

PRINT

```
PRINT("control_string"[, remaining_parameters])
```

This causes a single line to be inserted in the trace output file.

The *control_string* parameter is a character string defining the format of the line. It specifies literal text and sufficient format controls for the following parameters. Format controls should be separated by the underscore character '_', which becomes a blank in the output.

The format controls are:

%JW.nc

where:

% Is a token indicating a substitution format control.

J Is an optional control specifying either:

- + Right justified, the default.
- Left justified.

W Is an optional control that specifies the minimum field width. If no value is given, the PRINT function chooses a suitable value. If the value specified is too small, it is overridden with the value that the PRINT function would have used, had no value been specified.

n Is an optional control that specifies:

- For character strings, the maximum number of characters to be printed. If not specified, the default is all the characters in the string.
- For floating-point numbers, the number of digits to be printed to the right of the decimal point. If this number is not specified, the default is 1. This prevents a floating-point number being mistaken for an integer.

c Is a control character that *must* be specified. It determines the type of formatting performed. Possible values are:

B or b Binary format.

The corresponding parameter should be a fullword. Output has a single leading zero to minimize the binary string length.

D or d Decimal format.

The corresponding parameter should be a fullword.

E or e Exponential format.

The corresponding parameter should be in short form floating-point format.

F or f Floating-point format.

The corresponding parameter should be in short form floating-point format.

S or s Character string format.

The corresponding parameter should be a character string.

X or x Hexadecimal format.

The corresponding parameter should be a fullword.

The decimal point is only required if the optional *n* control is specified.

The *remaining_parameters* are the items that are to be formatted as directed by the control string and then sent to the trace file.

If there are more *remaining_parameters* than format controls, the excess parameters are ignored.

If there are more format controls than *remaining_parameters*, a single * is put in the trace file for each missing parameter.

Here are some examples of PRINT:

- PRINT("Trace_example") produces:

Trace example

- PRINT("%s","trace_example") produces:

trace example

- PRINT("%20s","trace_example") produces:

trace example

1---5---0---5---0---5---0 - (This line indicates where
the characters appear.)

- PRINT("%-20s","trace_example") produces:

trace example

1---5---0---5---0---5---0

- PRINT("%-20.10s","trace_example") produces:

trace example

1---5---0---5---0---5---0

- PRINT("Data_in_reg_%d_is_%x", 5, 5 GR) produces:

Data in register 5 is 02447E

if the data in general-purpose register 5 is X'02447E'.

Notes:

1. All spaces within a character string must be written as underscore (_) in the TRCESTR string.
2. Because they are within a TRCESTR statement, the *control_string*, and each of the *remaining_parameters* that is a character string, must be within either double quotation marks or a pair of single quotation marks.

STGREP

```
STGREP[({ONOFF|control})]
```

If set to ON, this specifies that a storage use report is to be included in the trace output.

The report is normally included in the trace file on termination or immediately before a GDDM-initiated abend.

If a *control* parameter with a numerical value greater than 1 is included in the statement, any storage report already started is included in the trace file immediately and a new storage report is started.

The *control* function is described on page 42.

TIME

```
TIME
```

This specifies that a time stamp is to be included in the trace output. The time stamp includes date and time. Time is shown in two formats: hh:mm:ss and the same value expressed in seconds.

TRACE

```
TRACE(number,ONOFF)
```

GDDM provides some special Trace Points that can be enabled by using the TRACE action function. When these trace points are enabled, information about specific areas of GDDM function will be traced.

Enabling trace points: Switch on tracing for the trace points you are interested in by using one or more TRCESTR statements, and using the TRACE action function. For example:

```
DEFAULT TRCESTR='TRACE(4,ON)'      CDPU specific tracing
DEFAULT TRCESTR='TRACE(11,ON)'     CGM specific tracing
```

Trace Points available: The following Trace points are available.

```

0 5080 Device information
4 CDPU specific information
8 Family 4 printing
11 CGM (Computer Graphics Metafile conversion,
    Queued Print Utility and DSCOPY information

```

5080IO

5080IO[({**ON**|OFF|control}[, *number*])]

If ON, this specifies that calls to the 5080 Graphics System workstation (using GDDM/graPHIGS) are to be traced. The format of the trace entries is the same as the format for parameter trace entries for calls to GDDM Base. The calls using GDDM/graPHIGS have GP as the first two letters of the call name.

The *number* parameter is optional. It specifies a limit for the number of bytes of data to be displayed. If this parameter is not specified, a default limit of 80 bytes is applied.

Event functions

Event functions, when used with an IF statement, restrict the amount of trace output produced by the action functions. That is, they qualify the conditions that, when satisfied, cause the specified trace actions.

The event functions that you can specify are summarized in this table:

Function	Syntax
ALL	ALL
Component	AIC CDU ICS IES IFSM IGKS ICU IMC IMD IMS ISE IVU INUM IPGR QPU TS IVSE
COUNT	(<i>number</i>)
ENTRY	ENTRY
EXIT	EXIT
Level	MOD SCMP NTIC MPNT API
LIMIT	LIMIT(<i>low_limit</i> [, <i>high_limit</i>])
NAME	NAME("module_name")
RANGE	RANGE(<i>low_event_number</i> [, <i>high_event_number</i>])

If you do not specify any event functions, the action functions are executed at the entry and exit of each module.

ALL

ALL

This specifies that all the actions specified by action functions are to be executed at the entry and exit of each module.

Component

```
AICICDUICSIIESIIFSMIGKSIICUIIMCIIMDIIMSIISEIIVUINUMIPGRIQPUITSIIVSE
```

This specifies the component to be traced. If you specify the module or subcomponent level, tracing will be done at that level within the component you specify with this parameter. If you do not specify a component, the default is ALL.

You can specify more than one component by putting an 'or' sign (I) between the codes for the components you want to trace. The components and their associated name prefixes are as follows:

Code	NAME	Component
AIC	ADMAxxxx	Application Program Interface Component
CDU	ADM4xxxx	Composite Document Print Utility
CSI	ADMYxxxx	Common Services Interface
ESI	ADMExxxx	Environmental Services Interface
FSM	ADMDxxxx	Full Screen Manager
GKS	ADMJxxxx	GDDM-GKS
ICU	ADMPxxxx	Interactive Chart Utility
IMC	ADM3xxxx	Image Manager Component
IMD	AEMxxxxx	GDDM-IMD
IMS	ADMKxxxx	Information Management System
ISE	ADMIxxxx	Image Symbol Editor
IVU	ADM5xxxx	GDDM-IVU
NUM	ADMNxxxx	Numerical Processing Routines
PGR	ADMBxxxx	Presentation Graphics Routines
QPU	ADMOxxxx	Queued Print Utility
TSI	ADMLxxxx	Terminal Services Interface
VSE	ADMVxxxx	Vector Symbol Editor.

You can use the NAME prefixes, given in the table, with the NAME function (described on page 41) to be more selective in your trace operation. For example, specifying NAME("ADMA*") is equivalent to specifying AIC.

COUNT

```
COUNT(number)
```

This specifies that trace records are to be produced every time the number of occurrences of the specified event matches the value of the *number* parameter. If you do not specify any event functions, the entry and exit of each module is considered to be a specified event.

In the following example, trace records are produced every fourth time module ADMEF* is entered:

```
[label] ADMMDF TRCESTR='IF ENTRY & NAME("ADMEF*") THEN      '
          ADMMDF TRCESTR='      IF COUNT(4) THEN FLOW          '
```

ENTRY

ENTRY

This specifies that tracing of entries to units at the specified level is required.

EXIT

EXIT

This specifies that tracing of exits from units at the specified level is required.

LevelMOD|SCMPNT|CMPNT|API

The following values specify, from high to low, the level at which tracing is to be done:

API	Application programming interface level
CMPNT	Component boundaries
SCMPNT	Subcomponent boundaries
MOD	Module boundaries.

MOD is the default value. The lower levels of tracing include the higher levels, so that MOD includes tracing at subcomponent, component, and API levels.

LIMITLIMIT(*low_limit*[, *high_limit*])

This specifies limits for tracing occurrences of a particular set of conditions. Trace records are produced every time the number of occurrences of the specified event lies in the range *low_limit* through *high_limit*. If you do not specify any event functions, the entry and exit of each module is considered to be a specified event.

The second parameter is optional; if it is omitted, a value of infinity is used.

In the following example, trace records are produced for the third, fourth, and fifth times that module ADMEF* is entered:

```
[label] ADMMDFT TRCESTR='IF ENTRY & NAME("ADMEF*") THEN      '
          ADMMDFT TRCESTR='      IF LIMIT(3,5) THEN FLOW      '
```

NAME

```
NAME("module_name")
```

The *module_name* parameter is a character string that either specifies a module name, or consists of the first part of a module name followed by an asterisk (*) to specify all modules whose names start with the characters given. For example, ADME* specifies ADMEAAA through ADMEZZZ.

Notes:

1. Only one * is allowed and it must be the last character in the string.
2. Because it is within a TRCESTR statement, the *module_name* must be within either double quotation marks or a pair of single quotation marks and it must be in uppercase.

The first part of each module name, with the component that contains it, is given in the table on page 39.

RANGE

```
RANGE(low_event_number[, high_event_number])
```

This specifies that trace records are to be produced for events with sequence numbers that lie in the range *low_event_number* through *high_event_number*.

Each module entry and exit is given an event sequence number; you can use these numbers to limit the trace output to the area that you are most interested in. If you omit the second parameter, a value of infinity is used.

Only those events that have numbers in the specified range, and meet the requirements of any action function specified, appear in the trace output.

For example, trace output produced by the statement:

```
[label] ADMMDFT TRCESTR='FLOW'
```

can include trace records numbered from 1 through 900. If you see that the problem you are examining is shown in records 210 through 230, you can limit future trace output to this area by using the statement:

```
[label] ADMMDFT TRCESTR='IF RANGE(190,250) THEN FLOW'
```

Note: You should broaden the range specified from just the area in which you are interested; trace record numbers may not correspond exactly from run to run because the modules may be loaded in a different order.

Control function

({**ON**|OFF|*control*})

Some of the action functions described above have an optional control function associated with them. You can specify one of the following for this function:

- ON** The trace interpreter takes the action specified. This is the default value for the control function.
- OFF** The trace interpreter does not take the action specified.
- control* You can specify a *control* relational expression. (The syntax of relational expressions is defined in Appendix D, “Trace-string syntax” on page 199.) The trace interpreter evaluates the expression:
 - If the value is zero, or logical false, the action specified is not taken. This is equivalent to OFF.
 - If the value is not zero, or logical true, the action specified is taken. This is equivalent to ON. For the STGREP action, values of greater than one cause a storage report to be produced immediately and a new storage report to be begun.

For example, *control* can be the name of an event function. The results of the following statements are the same:

```
[label] ADMMDFT TRCESTR='IF AIC OR FSM THEN STGREP(ON) ELSE STGREP(OFF) '
        ADMMDFT TRCESTR='STGREP(AIC|FSM) '
```

The TRCEWID keyword

Using the TRCEWID default keyword, you can specify the width of the trace output. The valid options are:

```
[label] ADMMDFT TRCEWID=SINGLE
        ADMMDFT TRCEWID=DOUBLE
```

The default is SINGLE, which produces 4-word hex output. DOUBLE produces 8-word hex output. This option reduces the amount of paper used.

_____ End of Product-sensitive programming interface _____

Format of trace output

Trace output begins with the trace control word and a GDDM build identification. Unless a partial transmission trace is the only option specified, a formatted listing of the GDDM defaults table is produced. Here is an explanation of some of the information contained in trace output.

Trace record type

A mnemonic shows the type of trace record:

CPNIN	Shows that a GDDM component has been entered to perform the function listed.
CPNOUT	Shows that a GDDM component has been exited after performing the function listed.
ERFIN	Shows that, on entry to a GDDM component, the standard GDDM error feedback areas contained error information as listed.
ERFOUT	Shows that, on leaving a GDDM component, the standard GDDM error feedback areas contained error information as listed.
SUBIN	Shows that the GDDM subcomponent listed has been entered.
SUBOUT	Shows that the GDDM subcomponent listed has been exited.
MODIN	Shows that the GDDM module listed has been entered.
MODOUT	Shows that the GDDM module listed has been exited.
MSGOUT	Is produced on leaving the message processor in GDDM. It contains the text of any message produced.

Trace explanation

The contents of the trace explanation vary depending on the trace record type, as follows:

CPNIN, CPNOUT

The trace explanation contains:

- The request control parameter (RCP) in mnemonic and hexadecimal format. The first byte of the RCP identifies the component being invoked, as follows:

X'00'	AIC	Application Interface Component
X'04'	TSI	Terminal Services Interface Component
X'08'	ESI	Environmental Services Interface Component
X'0C'	FSM	Full Screen Manager
X'10'	PGR	Presentation Graphics Routines
X'14'	ICU	Interactive Chart Utility
X'18'		Image Symbol Editor
X'1C'	CSI	Common Services Interface Component
X'20'		Vector Symbol Editor
X'24'	OPU	Output Print Utility
X'28'	IMS	IMS Interactive Utility Scheduler
X'2C'	NMR	Numerical Processing Routines
X'30'	IMD	GDDM-IMD
X'38'	GKS	GDDM-GKS
X'3C'	IMC	Image Manager Component
X'40'	CDU	Composite Document Print Utility
X'44'	IVU	GDDM-IVU

- An interpretation of the RCP.
- For calls to the Environmental Services Interface (ESI), for storage, program load/delete, or abend processing, an interpretation of the parameters to those services.

ERFIN, ERFOUT

The trace explanation contains:

- The associated error message number, severity, and text. The string '(INTERNAL)' is produced for error codes that are purely internal to GDDM.

SUBIN, SUBOUT, MODIN, MODOUT

The trace explanation contains:

- The module identification, including the module name.
- For SUBOUT or MODOUT, the value of register 15, if it appears to be a return code.

Examples of trace output from TRCESTR statements

The following pages show examples of output obtained from various sets of trace statements. In each case, the statements used are given.

The output from each trace begins with a listing of the default table. This is included in the first example, but thereafter is omitted. Other records may also be omitted, for the sake of clarity. Where records are omitted, this is shown.

Example 1. Tracing a range of events

To produce normal trace output records for events 205 through 215, the following statement was used:

```
[label] ADMMDFTR TRCESTR='IF RANGE(205,215) THEN FLOW'
```

The event sequence numbers are shown in the following output.

Note: A description of ADMTDFT is given in "ADMTDFT – general defaults table" on page 151.

1GDDM DIAGNOSTIC TRACE - ORIGIN HPWJ

07 AUG 1992 8:41 AM

TRACE WORD = '00000000'X BUILDID = 92214

ADMTDFT - GENERAL DEFAULTS TABLE COMMON SECTION

006E9180	C4C6E340	000001F8	00000001	00000000	*DFT ...8.....*
006E9190	00000000	00000000	00000000	00000000	*.....*
006E91A0	00000000	00000108	00000168	00000180	*.....*
006E91B0	00130000	C1040101	7F400000	00000004	*....A...".....*
006E91C0	00000002	007B5750	000000A4	00000000	*....#.&...u....*
006E91D0	00000064	00000000	00000600	00000400	*.....*
006E91E0	00002000	00000008	007D73A8	006F7F78	*.....'.y.?'..*
006E91F0	00000000	00000000	00000004	00000000	*.....*
006E9200	00000000	00000000	C1C6E3C3	F0F3F9F5	*.....AFTC0395*
006E9210	5C404040	40404040	5C404040	40404040	** * *
006E9220	5C404040	40404040	5C404040	40404040	** * *
006E9230	5C404040	40404040	5C404040	40404040	** * *
006E9240	5C404040	40404040	5C404040	40404040	** * *
006E9250	00000000	006F00B8	50050000	00000000	*.....?..&.....*
006E9260	00000000	00000000	00000025	0000015F	*.....~*
006E9270	80000000	00000000	C1C4D4C9	D2404040	*.....ADMIK *
006E9280	C1C4D4E5	D2404040			*ADMVK *
- OBJECT TYPE DEFAULTS					
006E9288			C1C4D4E2	E8D4C2D3	* ADMSYMBL*

```

006E9290 C1C4D4C7 C7D4C1D7 C1C4D4E2 C1E5C540 *ADMGGMAPADMSAVE *
006E92A0 C1C4D4C3 C6D6D9D4 C1C4D4C3 C4C1E3C1 *ADMCFORMADMCDATA*
006E92B0 C1C4D4E3 E4E3D7C7 C1C4D4C7 C4C64040 *ADMTUTPGADMGDF *
006E92C0 C1C4D4C7 D2E2D440 C1C4D4C3 C4C5C640 *ADMGKSM ADMCDEF *
006E92D0 C1C4D4D7 D9D6D140 C1C4D4C9 D4C74040 *ADMPROJ ADMIMG *
006E92E0 C1C4D4D7 C3404040 *ADMPIC *
- USER EXIT DEFAULTS
006E92E8 00000000 00000BBB 00000000 * .....*
006E92F0 00000000 00000BBB 00000000 *.....*
- SUBSYSTEM SPECIFIC DEFAULTS
006E9300 80000000 00000000 C1C4D4E4 E3F14040 *.....ADMUT1 *
006E9310 C1C4D4C4 C5C3D240 C1C4D4D7 D9C9D5E3 *ADMDECK ADMPRINT*
006E9320 C1C4D4D3 C9E2E340 C3D6D7E8 40404040 *ADMLIST COPY *
006E9330 C1C4D4C9 C6D4E340 C1C4D4D4 E2D34040 *ADMIFMT ADMMSL *
006E9340 C1C4D4C9 D4C1C7C5 C1C4D4C3 D6D34E40 *ADMIMAGEADMCOL+ *
006E9350 C5E7C1D4 D7D3C5F1 C1C4D4E3 D9C1C3C5 *EXAMPLE1ADMTRACE*
006E9360 D7D9D6C6 C9D3C540 C1C4D4C4 C5C6E240 *PROFILE ADMDEFS *
006E9370 C1C4D4C3 C7D44040 *ADMCGM *
ADMTNICK - NICKNAME LIST ENTRY
006F7F78 D5C9C3D2 0000006C * NICK...%*
006F7F80 006E9A90 00000000 60000000 006F7FBC *.>.....-....? ".*
006F7F90 006F7FBC 006F7FBC 006F7FD0 00000000 *."..?"..?" }....*
006F7FA0 00000000 00000000 00000000 40404040 *.....*
006F7FB0 40404040 00000005 00000000 0000001C * .....*
006F7FC0 00000001 0000001D 00000001 00000001 *.....*
006F7FD0 00000000 00000000 00000000 00000000 *.....*
006F7FE0 00000000 *....*
ADMTNICK - NICKNAME LIST ENTRY
006E9A90 D5C9C3D2 0000005C 006F0038 006F7F78 *NICK...*.?....? ".*
006E9AA0 40000000 006E9AD4 006E9ADC 006E9AEC * ....>.M.>...>..*
006E9AB0 006E9AEC 00000000 00000001 00000001 *.>.....*
006E9AC0 00000002 40404040 40404040 00000000 *....*
006E9AD0 00000000 D7C3D7D9 C9D5E340 5C404040 *....PCPRINT * *
006E9AE0 40404040 C1C4D4D7 C3D7D9E3 * ADMPCPRT *
ADMTNICK - NICKNAME LIST ENTRY
006F0038 D5C9C3D2 0000005C * NICK...**
006F0040 00000000 006E9A90 40000000 006F007C *.....>.. ....?..@*
006F0050 006F0084 006F0094 006F0094 00000000 *?.d.?.m.?.m....*
006F0060 00000001 00000001 00000002 40404040 *.....*
006F0070 40404040 00000000 00000000 D6E2F2D7 * .....OS2P*
006F0080 D9C9D5E3 5C404040 40404040 C1C4D4D7 *RINT* ADMP*
006F0090 D4D6D740 *MOP *

```

DEFAULT TRCESTR='IF RANGE(205,215) THEN FLOW '

```

00000205 05 MODOUT - - ADMERO0V 96067 V3R2.0
CPNOUT - - ESSGLU ('080C0410'X) - UNCOND GET LOCAL STG (000464 AT '00649C18'X)
00000206 05 CPNIN - - LRLSE ('04000700'X) - RELEASE I/B BUFFER
MODIN - - ADMLR01V 92214 V3R2.0
00000207 06 MODIN - - -ADMLR11V 92214 V3R2.0
00000208 07 CPNIN - - - ESSFI ('080C0840'X) - FREE INSTANCE STG (001536 AT '006BD5D8'X)
MODIN - - - ADMERO0V 92217 V3R2.0
00000209 07 MODOUT - - - ADMERO0V 92217 V3R2.0
CPNOUT - - - ESSFI ('080C0840'X) - FREE INSTANCE STG
00000210 06 MODOUT - - -ADMLR11V 92214 V3R2.0
00000211 05 MODOUT - - ADMLR01V 92214 V3R2.0
CPNOUT - - LRLSE ('04000700'X) - RELEASE I/B BUFFER
00000212 05 CPNIN - - ABLIMG ('00010102'X) - BUILD INT. ERROR MSG.
MODIN - - ADMACFP 92217 V3R2.0
00000213 06 SUBIN - - -ADMAEP 92214 V3R2.0
00000214 07 MODIN - - - ADMAEMP 92214 V3R2.0
00000215 08 CPNIN - - - -EPSLGC ('08040488'X) - COND GROUP LOAD (ADMAET3A)
MODIN - - - -ADMERO0V 92217 V3R2.0

```

00001391 END OF GDDM DIAGNOSTIC TRACE 07 AUG 1992 8:41 AM

Example 2. Restricting the level at which tracing takes place

Here is the statement for example 2:

```
[label] ADMMDFT TRCESTR='STGREP IF API THEN DO PARMSF(32000) END'
```

This statement requests that a storage use report and call parameters (in formatted notation) are to be included in the trace output. The number 32000 specified in the PARMSF parameter causes up to 32000 bytes to be listed for each traced parameter. (If no number is specified, the default value of 80 bytes is used.) Tracing is at the application programming interface (API) level.

The statement produced the following output:

1GDDM DIAGNOSTIC TRACE - ORIGIN MOYNIHAN 07 MAR 1996 4:04 PM

TRACE WORD = '00000000'X BUILDID = 96067

```
*****
*
* Trace defaults and nickname table omitted from this example
* for clarity
*
*****
```

```
DEFAULT TRCESTR='STGREP IF API THEN DO PARMSF(32000) END'
00000001 01 CPNIN FSINIT ('0C000001'X) - INITIALISATION
00000014 01 CPNOUT FSINIT ('0C000001'X) - INITIALISATION
```

```
00000015 01 CPNIN GSWIN ('0C0C0002'X) - SPECIFY WINDOW
PTRACE 1 FLOAT 0
PTRACE 2 FLOAT 100
PTRACE 3 FLOAT 0
PTRACE 4 FLOAT 100
```

```
00000337 01 CPNOUT GSWIN ('0C0C0002'X) - SPECIFY WINDOW
PTRACE 1 FLOAT ---INPUT ONLY PARAMETER-----
PTRACE 2 FLOAT ---INPUT ONLY PARAMETER-----
PTRACE 3 FLOAT ---INPUT ONLY PARAMETER-----
PTRACE 4 FLOAT ---INPUT ONLY PARAMETER-----
```

```
00000338 01 CPNIN GSCOL ('0C0C0701'X) - SET COLOR
PTRACE 1 FIXED 1
00000375 01 CPNOUT GSCOL ('0C0C0701'X) - SET COLOR
PTRACE 1 FIXED ---INPUT ONLY PARAMETER-----
```

```
00000376 01 CPNIN GSMOVE ('0C0C0400'X) - MOVE TO
PTRACE 1 FLOAT 0
PTRACE 2 FLOAT 0
00000379 01 CPNOUT GSMOVE ('0C0C0400'X) - MOVE TO
PTRACE 1 FLOAT ---INPUT ONLY PARAMETER-----
PTRACE 2 FLOAT ---INPUT ONLY PARAMETER-----
```

```
00000380 01 CPNIN GSLINE ('0C0C0401'X) - LINE TO
PTRACE 1 FLOAT 100
PTRACE 2 FLOAT 0
00000389 01 CPNOUT GSLINE ('0C0C0401'X) - LINE TO
PTRACE 1 FLOAT ---INPUT ONLY PARAMETER-----
PTRACE 2 FLOAT ---INPUT ONLY PARAMETER-----
```

```
00000390 01 CPNIN GSLINE ('0C0C0401'X) - LINE TO
```

```

PTRACE    1 FLOAT                100
PTRACE    2 FLOAT                100
00000395 01 CPNOUT  GSLINE ('0C0C0401'X) - LINE TO
PTRACE    1 FLOAT  ---INPUT ONLY PARAMETER-----
PTRACE    2 FLOAT  ---INPUT ONLY PARAMETER-----

00000396 01 CPNIN   GSLINE ('0C0C0401'X) - LINE TO
PTRACE    1 FLOAT                0
PTRACE    2 FLOAT                100
00000401 01 CPNOUT  GSLINE ('0C0C0401'X) - LINE TO
PTRACE    1 FLOAT  ---INPUT ONLY PARAMETER-----
PTRACE    2 FLOAT  ---INPUT ONLY PARAMETER-----

00000402 01 CPNIN   GSLINE ('0C0C0401'X) - LINE TO
PTRACE    1 FLOAT                0
PTRACE    2 FLOAT                0
00000407 01 CPNOUT  GSLINE ('0C0C0401'X) - LINE TO
PTRACE    1 FLOAT  ---INPUT ONLY PARAMETER-----
PTRACE    2 FLOAT  ---INPUT ONLY PARAMETER-----

00000408 01 CPNIN   ASDFLD ('0C080700'X) - DEFINE ALPHA FIELD
PTRACE    1 FIXED                1
PTRACE    2 FIXED                5
PTRACE    3 FIXED                5
PTRACE    4 FIXED                1
PTRACE    5 FIXED                25
PTRACE    6 FIXED                0
00000433 01 CPNOUT  ASDFLD ('0C080700'X) - DEFINE ALPHA FIELD
PTRACE    1 FIXED  ---INPUT ONLY PARAMETER-----
PTRACE    2 FIXED  ---INPUT ONLY PARAMETER-----
PTRACE    3 FIXED  ---INPUT ONLY PARAMETER-----
PTRACE    4 FIXED  ---INPUT ONLY PARAMETER-----
PTRACE    5 FIXED  ---INPUT ONLY PARAMETER-----
PTRACE    6 FIXED  ---INPUT ONLY PARAMETER-----

00000434 01 CPNIN   ASCPUT ('0C080603'X) - SET CHARACTER CODES
PTRACE    1 FIXED                1
PTRACE    2 DIM                 25
PTRACE    3 CHAR  'GDDM TRACE DEMO ' C7C4C4D4 40E3D9C1 C3C540C4 C5D4D600
PTRACE    3 CHAR  ' ' 00000000 00000000 00
00000441 01 CPNOUT  ASCPUT ('0C080603'X) - SET CHARACTER CODES
PTRACE    1 FIXED  ---INPUT ONLY PARAMETER-----
PTRACE    2 DIM    ---INPUT ONLY PARAMETER-----
PTRACE    3 CHAR   ---INPUT ONLY PARAMETER-----

00000442 01 CPNIN   ASREAD ('0C100000'X) - READ
PTRACE    1 FIXED  ---OUTPUT ONLY PARAMETER-----
PTRACE    2 FIXED  ---OUTPUT ONLY PARAMETER-----
PTRACE    3 FIXED  ---OUTPUT ONLY PARAMETER-----
00001085 01 CPNOUT  ASREAD ('0C100000'X) - READ
PTRACE    1 FIXED                0
PTRACE    2 FIXED                0
PTRACE    3 FIXED                0

00001086 01 CPNIN   FSTERM ('0C000000'X) - TERMINATION
00001461 01 CPNOUT  FSTERM ('0C000000'X) - TERMINATION

```

STORAGE USE REPORT

LENGTH	AT ADDRESS	GOT BY	(SEQ)	FREED BY	(SEQ)
000212	'005D89F8'X	ADMDSRO	00000012	ADMDSRO	00001450
000008	'005D8FE0'X	ADMDSDS	00000023	ADMDSDS	00000188
000624	'005D7348'X	ADMDSDS	00000027	ADMDSDS	00001447

000696	'005D75B8'	X	ADMLIN1V	00000039	ADMLTM1V	00001433	
000028	'005D83E0'	X	ADMYRSRL	00000043	ADMYRSRL	00001429	
001536	'0056A000'	X	ADMLAC1V	00000049	ADMLRL1V	00000076	
000172	'005D8AD0'	X	ADMDSDS	00000069	ADMDSDS	00001445	
001024	'005D8400'	X	ADMER00V	00000072	ADMER00V	00001404	(SMALL STORAGE POOL)
000232	'005D8408'	X	ADMDSDS	00000073	ADMDSDS	00000186	
000390	'005D84F0'	X	ADMDSF1	00000094	ADMDSCL	00001403	
000094	'005D8B80'	X	ADMDSII	00000097	ADMDSII	00001396	
000069	'005D8678'	X	ADMDSII	00000099	ADMDSII	00001398	
001024	'0056A000'	X	ADMER00V	00000103	ADMER00V	00001412	(SMALL STORAGE POOL)
000412	'0056A008'	X	ADMDSF1	00000104	ADMDSCL	00001411	
002000	'0056A830'	X	ADMDSQL	00000113	ADMDSQL	00000114	
000156	'005D86C0'	X	ADMDSQP	00000119	ADMDSQP	00000120	
000684	'005D7870'	X	ADMDSQ	00000129	ADMDSCL	00001407	
000684	'005D7B20'	X	ADMDSQ	00000131	ADMDSCL	00001409	
000222	'005D86C0'	X	ADMWINT	00000146	ADMWTRM	00001300	
000040	'005D8950'	X	ADMDOOC	00000160	ADMDOOC	00001094	
000176	'0056A1A8'	X	ADMDNCRT	00000176	ADMDNDEL	00001254	
000176	'0056A258'	X	ADMDNCRT	00000178	ADMDNDEL	00001246	
000176	'0056A308'	X	ADMDNCRT	00000180	ADMDNDEL	00001248	
000092	'005D87A0'	X	ADMDNCRT	00000182	ADMDNDEL	00001252	
000068	'0056A3B8'	X	ADMDBCRT	00000205	ADMDBDEL	00001240	
000072	'005D8408'	X	ADMDECRT	00000221	ADMDEDEL	00001236	
004096	'00569000'	X	ADMER00V	00000237	ADMER00V	00001232	(SMALL STORAGE POOL)
000717	'00569008'	X	ADMDCPC	00000238	ADMDCPC	00001231	
001024	'0056AC00'	X	ADMER00V	00000281	ADMER00V	00001322	(SMALL STORAGE POOL)
000260	'0056AC08'	X	ADMDGI1	00000282	ADMDGI1	00001321	
000420	'0056AD10'	X	ADMDGWI	00000294	ADMDGWI	00001214	
000124	'005D8450'	X	ADMDGS1	00000298	ADMDGS1	00001206	
004080	'00568000'	X	ADMYGQC	00000302	ADMYGQC	00001202	
002024	'0056A418'	X	ADMDGPAX	00000310	ADMDGPAX	00001174	
001024	'00567000'	X	ADMER00V	00000315	ADMER00V	00001193	(SMALL STORAGE POOL)
000272	'00567008'	X	ADMYGQC	00000316	ADMYGQC	00001170	
000268	'0056AEB8'	X	ADMDGSE	00000323	ADMDGSE	00001163	
004080	'00564000'	X	ADMYGQC	00000331	ADMYGQC	00001150	
001544	'005679F8'	X	ADMDGS1	00000366	ADMDGS1	00001075	
001552	'00563000'	X	ADMYGQC	00000425	ADMYGQC	00001227	
002560	'00561000'	X	ADMDASCR	00000429	ADMDCPC	00001223	
001536	'00563A00'	X	ADMLAC1V	00000466	ADMLRL1V	00001062	
001024	'005675F8'	X	ADMER00V	00000482	ADMER00V	00001390	(SMALL STORAGE POOL)
000248	'00567600'	X	ADMDHIN	00000483	ADMDSCL	00001389	
000512	'005676F8'	X	ADMDHIN	00000485	ADMDSCL	00001387	
000024	'005D84D0'	X	ADMDHIN	00000487	ADMDHDF	00001196	
000256	'00567118'	X	ADMDHCO	00000490	ADMDHDF	00001192	
004096	'00560000'	X	ADMER00V	00000493	ADMER00V	00001189	(SMALL STORAGE POOL)
001024	'00560008'	X	ADMDHCO	00000494	ADMDHDF	00001188	
005120	'0055E000'	X	ADMDHCO	00000496	ADMDHSS	00001183	
004560	'0055B000'	X	ADMDHUW	00000517	ADMDCES	00001353	
002572	'0055C5F0'	X	ADMDXRIO	00001002	ADMDNDEL	00001250	
001536	'00561A00'	X	ADMDDSGO	00001010	ADMDDSGO	00001025	
001546	'0055F9F0'	X	ADMDDSGO	00001012	ADMDDSGO	00001027	
000128	'005D8E00'	X			ADMATPE	00001472	
000108	'005D8E80'	X			ADMATPE	00001474	
000108	'005D8EF0'	X			ADMATPE	00001476	
000068	'005D8398'	X			ADMATPE	00001478	
000128	'005D8F60'	X			ADMATPE	00001480	
000048	'005D8808'	X			ADMATPE	00001482	
000828	'005D7008'	X			ADMATPE	00001484	
004096	'005D7000'	X			ADMER00V	00001485	
000278	'005D8838'	X			ADMATPE	00001488	
000048	'005D8368'	X			ADMACUP	00001491	

00001527 END OF GDDM DIAGNOSTIC TRACE 07 MAR 1996 4:04 PM

Example 3. Requesting a full transmission trace

The statement for example 3 is:

```
[label] ADMMDFT TRCESTR='FULLIO'
```

The trace output is:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN HPWJ                07 MAR 1996  8:47 AM

TRACE WORD = '00000000'X   BUILDID = 96067

*****
*
* Trace defaults and nickname table omitted from this example
* for clarity
*
*****

DEFAULT TRCESTR='FULLIO '

OUTBOUND TRANSMISSION SENT

I/O BUFFER                                LENGTH = 00002    TIME 08:47:13
006BD018                                7EC2                *      =B      *

OUTBOUND TRANSMISSION SENT

I/O BUFFER                                LENGTH = 00006    TIME 08:47:13
006BD018                                F3000501 FF02        *      3..... *

INBOUND TRANSMISSION RECEIVED

I/O BUFFER                                LENGTH = 00278    TIME 08:47:13
006BD018                                88001781 81010000    *      h..aa...*
006BD020      50002001 03E80E26 03E80E26 0A120A00    *&....Y...Y.....*
006BD030      001B8185 82000A12 00000000 07000000    *..aeb.....*
006BD040      02B90025 0100F103 C3013600 1181A600    *.....l.C....aw.*
006BD050      000B0100 00500018 00500020 00168186    *.....&...&....af*
006BD060      000800F4 F1F1F2F2 F3F3F4F4 F5F5F6F6    *...4112233445566*
006BD070      F7F7000D 81870400 F0F1F1F2 F2F4F400    *77..ag..0112244.*
006BD080      07818800 01020006 81990000 0013819E    *.ah.....ar....a.*
006BD090      80000300 0E000104 01810104 03000300    *.....a.....*
006BD0A0      1D819500 0003000E 00010104 01810209    *.an.....a...*
006BD0B0      02D7C3D3 D2011080 04030003 0021819E    *.PCLK.....a.*
006BD0C0      80000300 0E000204 01810304 0300030E    *.....a.....*
006BD0D0      0202B901 F45C5CC7 D8C6C4C5 C6002181    *....4**GQFDEF..a*
006BD0E0      9E800003 000E0002 04018104 04030003    *.....a.....*
006BD0F0      0E0202B9 01F4C9C2 D4F4F0F1 F9A50021    *.....4IBM4019v..*
006BD100      819E8000 03000E00 02040181 05040300    *a.....a.....*
006BD110      030E0202 B901F4F4 F0F1F9D8 40404000    *.....44019Q   .*
006BD120      0F818080 81858687 8895999E A6A8      *.a..aefghnr.wy  *

OUTBOUND TRANSMISSION SENT

I/O BUFFER                                LENGTH = 00002    TIME 08:47:14
00666018                                F1C2                *      1B      *

OUTBOUND TRANSMISSION SENT

I/O BUFFER                                LENGTH = 00008    TIME 08:47:14
00666018                                F3000701 FF0300A8    *      3.....y*
```

tracing

I/O BUFFER	LENGTH = 00010	TIME 08:47:14
00666018	88000981 A80200F0	* h..ay..0*
00666020 FFFF		*.. *

OUTBOUND TRANSMISSION SENT

I/O BUFFER	LENGTH = 00034	TIME 08:47:15
00666018	F3000403 8000080F	* 3.....*
00666020 02000081 01000F10	13000003 08015800	*...a.....*
00666030 00010000 000601FF	0380	*..... *

INBOUND TRANSMISSION RECEIVED

I/O BUFFER	LENGTH = 00488	TIME 08:47:15
00666018	8800080F 02000081	* h.....a*
00666020 01000F10 13000003	08011800 5B010000	*.....\$...*
00666030 00C1819D 00020000	00000301 000F0119	*.Aa.....*
00666040 22066800 4751464E	4C555320 0F011922	*.....+<.....*
00666050 06680047 51464E48	5553200F 01192206	*.....+.....*
00666060 68004751 464E4255	53200F01 19220667	*.....+.....*
00666070 00475146 46494C45	530F0119 22066700	*.....<.....*
00666080 47514655 50442020	0F011922 12400047	*....&..... *
00666090 51465359 5320200F	01192206 67004751	*.....*
006660A0 46464F4E 20200F01	19220668 00475146	*.. +.....*
006660B0 54524420 200F0119	22066700 47514654	*.....*
006660C0 52432020 0F011922	06660047 51464C45	*.....<..*
006660D0 5620200F 01192206	67004751 46434C4E	*.....<+*
006660E0 20200F01 19220667	00475146 4E595553	*.....+...*
006660F0 20001B81 81010000	50002001 03E80E26	*...aa...&...Y..*
00666100 03E80E26 0A120A00	00000000 006D81B4	*.Y....._a.*
00666110 000400FF FF00FF00	100000FF FFFF0000	*.....*
00666120 0100FF00 000002FF	00000000 03FFFF00	*.....*
00666130 00000400 00FF0000	0500FFFF 000006FF	*.....*
00666140 00FF0000 07000000	00000883 83830000	*.....ccc..*
00666150 0900AA00 00000AAA	00000000 0BAAAA00	*.....*
00666160 00000C00 00920000	0D00AA92 00000E80	*.....k.....k....*
00666170 00920000 0FCDCDCD	00001B81 B600000A	*.k.....a....*
00666180 12000000 10010000	F0020200 00000000	*.....0.....*
00666190 02B90025 001881B2	00090007 01010202	*.....a.....*
006661A0 03030404 05050606	07070808 001081B0	*.....a....*
006661B0 00020000 00FFFF05	02004000 001181B3	*.....a....*
006661C0 00030290 09010003	0180007F FF000F81	*....."....a*
006661D0 B3000408 40070300	03000002 000F81B3	*....a....*
006661E0 00070840 07030001	00001B00 1581B100	*...a....*
006661F0 00000000 FFFF0600	06000601 FFFFFFFF8	*.....8*

INBOUND CONTROL SDP : REQUEST FRAME

006BD6A8	08011800 5B010000	*\$...*
----------	-------------------	--------------

MERGED QUERY REPLY

006D9BB8	001B8181 01000050	* ..aa...admd2tr2
006D9BC0 00200103 E80E2603	E80E260A 120A0000	*....Y...Y.....*
006D9BD0 00000000 1B818582	000A1200 00000007	*.....aeb.....*
006D9BE0 00000002 B9002501	00F103C3 01360011	*.....1.C....*
006D9BF0 81A60000 0B010000	50001800 50002000	*aw.....&...&...*
006D9C00 16818600 0800F4F1	F1F2F2F3 F3F4F4F5	*.af...4112233445*
006D9C10 F5F6F6F7 F7000D81	870400F0 F1F1F2F2	*56677..ag..01122*
006D9C20 F4F40007 81880001	02000681 99000000	*44..ah.....ar...*
006D9C30 13819E80 0003000E	00010401 81010403	*.a.....a....*
006D9C40 0003001D 81950000	03000E00 01010401	*....an.....*
006D9C50 81020902 D7C3D3D2	01108004 03000300	*a...PCLK.....*


```

006D9C60 21819E80 0003000E 00020401 81030403 *.a.....a...*
006D9C70 00030E02 02B901F4 5C5CC7D8 C6C4C5C6 *.....4**GQFDEF*
006D9C80 0021819E 80000300 0E000204 01810404 *..a.....a...*
006D9C90 0300030E 0202B901 F4C9C2D4 F4F0F1F9 *.....4IBM4019*
006D9CA0 A5002181 9E800003 000E0002 04018105 *v..a.....a.*
006D9CB0 04030003 0E0202B9 01F4F4F0 F1F9D840 *.....44019Q *
006D9CC0 4040000F 81808081 85868788 95999EA6 * ..a..aefghnr.w*
006D9CD0 A8000981 A80200F0 FFFF00C1 819D0002 *y..ay..0...Aa...*
006D9CE0 00000000 0301000F 01192206 68004751 *.....*
006D9CF0 464E4C55 53200F01 19220668 00475146 *.+<.....*
006D9D00 4E485553 200F0119 22066800 4751464E *+.....+*
006D9D10 42555320 0F011922 06670047 51464649 *.....*
006D9D20 4C45530F 01192206 67004751 46555044 *<.....&.*
006D9D30 20200F01 19221240 00475146 53595320 *.....*
006D9D40 200F0119 22066700 47514646 4F4E2020 *.....|+.*
006D9D50 0F011922 06680047 51465452 4420200F *.....*
006D9D60 01192206 67004751 46545243 20200F01 *.....*
006D9D70 19220666 00475146 4C455620 200F0119 *.....<.....*
006D9D80 22066700 47514643 4C4E2020 0F011922 *.....<+.....*
006D9D90 06670047 51464E59 55532000 6D81B400 *.....+....._a...*
006D9DA0 0400FFFF 00FF0010 0000FFFF FF000001 *.....*
006D9DB0 00FF0000 0002FF00 00000003 FFFF0000 *.....*
006D9DC0 00040000 FF000005 00FFFF00 0006FF00 *.....*
006D9DD0 FF000007 00000000 00088383 83000009 *.....ccc...*
006D9DE0 00AA0000 000AAA00 0000000B AAAA0000 *.....*
006D9DF0 000C0000 9200000D 00AA9200 000E8000 *...k.....k....*
006D9E00 9200000F CDCDCD00 001B81B6 00000A12 *k.....a.....*
006D9E10 00000010 010000F0 02020000 00000002 *.....0.....*
006D9E20 B9002500 1881B200 09000701 01020203 *.....a.....*
006D9E30 03040405 05060607 07080800 1081B000 *.....a...*
006D9E40 02000000 FFFF0502 00400000 1181B300 *.....a...*
006D9E50 03029009 01000301 80007FFF 000F81B3 *....."....a.*
006D9E60 00040840 07030003 00000200 0F81B300 *... ..a...*
006D9E70 07084007 03000100 001B0015 81B10000 *.. ..a...*
006D9E80 000000FF FF060006 000601FF FFFF8 *.....8 *

```

PREPARING PAGE FOR PARTITION

```

006DAF9C                                00000000 *          ....*

```

OUTBOUND TRANSMISSION SENT

```

I/O BUFFER                                LENGTH = 00005      TIME 08:47:16
005E93D0    F3000403 80                                *3....      *

```

OUTBOUND TRANSMISSION SENT

```

I/O BUFFER                                LENGTH = 00057      TIME 08:47:17
005E93D0    F300080F 02000081 01000F10 13000003 *3.....a.....*
005E93E0    08011800 5B020000 00210F10 00C00070 *...$......{...*
005E93F0    0C000000 00540000 0C000000 00C104FE *.....A...*
005E9400    70FEE081 04018F01 1F                                *..\.a.....*

```

INBOUND TRANSMISSION RECEIVED

```

I/O BUFFER                                LENGTH = 00024      TIME 08:47:17
005E93D0    8800080F 02000081 01000F10 13000003 *h.....a.....*
005E93E0    08019800 5B020000                                *..q.$...      *

```

INBOUND CONTROL SDP : POSITIVE RESPONSE

```

006BD6A8                                08019800 5B020000 *          ..q.$...*

```

OUTBOUND TRANSMISSION SENT

```

I/O BUFFER                                LENGTH = 00016    TIME 08:47:18
005E93D0    F3000F40    00F1F311    09FF1D60    11000013    *3... .13....-....*

INBOUND TRANSMISSION RECEIVED

I/O BUFFER                                LENGTH = 00003    TIME 08:47:18
005E93D0    F34040                                *3                                *

00001399 END OF GDDM DIAGNOSTIC TRACE  07 MAR 1996  8:47 AM

```

Example 4. Including the contents of registers in your trace

You may find it helpful to include the contents of the general-purpose registers (GREGS), or the floating-point registers (FREGS), or both, in your trace output like this:

```
[label] ADMMDFTRCESTR='IF COUNT(100) THEN FLOW GREGS FREGS'
```

The trace output for this statement is:

```

1GDDM DIAGNOSTIC TRACE - ORIGIN HPWJ                                07 MAR 1996  8:48 AM

TRACE WORD = '00000000'X    BUILDID = 96067

*****
*                                                                    *
* Trace defaults and nickname table omitted from this example      *
* for clarity                                                        *
*                                                                    *
*****

DEFAULT TRCESTR='IF COUNT(100) THEN FLOW GREGS FREGS '

00000100 03 CPNIN  - EPSLGU ('08040480'X) - UNCOND GROUP LOAD      (ADMDSF1 )
              MODIN  - ADMER00V 96067 V3R2.0

GENERAL PURPOSE REGISTERS
GPR0      = 006BDF A5 006E8960 006BDD B0 006E8A09
GPR4      = 006BDD B8 000C7874 006BD038 006E8000
GPR8      = 006E8640 006D9398 006DAFA8 400C5D98
GPR12     = 000C6D97 006E8858 400C67E4 800724B8
FLOATING POINT REGISTERS
FPR0      = 00000000 00000000                0                0
FPR2      = 42640000 00000000 1.000000E+02  1.0000000E+02
FPR4      = 00000000 00000000                0                0
FPR6      = 00000000 00000000                0                0
00000200 06 MODOUT - -ADMLR01V 96067 V3R2.0
              CPNOUT - -LCONV ('04000500'X) - CONVERSE

GENERAL PURPOSE REGISTERS
GPR0      = 006E9604 006E982C 006D9688 000E0851
GPR4      = 006D9690 006E9610 00000001 00000000
GPR8      = 000EC480 006D9398 000EA482 000EB481
GPR12     = 000EC163 006E97A8 500EA896 800CE000
FLOATING POINT REGISTERS
FPR0      = 00000000 00000000                0                0
FPR2      = 42640000 00000000 1.000000E+02  1.0000000E+02
FPR4      = 00000000 00000000                0                0
FPR6      = 00000000 00000000                0                0
00000300 15 CPNIN  - - - - ESSFI ('080C0840'X) - FREE INSTANCE STG (000298 AT '006497F8'X)
              MODIN  - - - - ADMER00V 96067 V3R2.0

GENERAL PURPOSE REGISTERS
GPR0      = 00649922 00667C54 00002133 00667C70
GPR4      = 000A4000 00000003 00667C68 006E9EA8

```

```

GPR8      = 006497F8 006E8000 000A5029 600A402A
GPR12     = 000A56CA 00667AC0 400A5598 800724B8
FLOATING POINT REGISTERS
FPR0      = 00000000 00000000 0 0
FPR2      = 42640000 00000000 1.000000E+02 1.0000000E+02
FPR4      = 00000000 00000000 0 0
FPR6      = 00000000 00000000 0 0
00000400 05 SUBIN - - ADMNCRT 96067 V3R2.0

GENERAL PURPOSE REGISTERS
GPR0      = 00000A00 000DAEFC 0000000C 0C2C0000
GPR4      = 000DAEFC 00000030 00649C18 000C7A05
GPR8      = 0014900C 006D9398 006BDDb0 400CC9A0
GPR12     = 000DA027 006E8CD8 400CCC88 80149088
FLOATING POINT REGISTERS
FPR0      = 43E25FFE F0000000 3.622000E+03 3.6220000E+03
FPR2      = 41100000 F0000000 1.000000E+00 1.0000000E+00
FPR4      = 00000000 00000000 0 0
FPR6      = 00000000 00000000 0 0
00000500 12 MODOUT - - - - -ADMYR000 96067 V3R2.0
              CPNOUT - - - - -YSCNMP ('1C100000'X) - SCAN MODULE MAP

GENERAL PURPOSE REGISTERS
GPR0      = 00000001 006F53FC 00000001 00000004
GPR4      = 006E9D78 006E9D74 006F513B 006F5420
GPR8      = 006F5424 006D9398 006E8000 5007424A
GPR12     = 00000018 006F5390 7007443A 80079808
FLOATING POINT REGISTERS
FPR0      = 43E25FFE F0000000 3.622000E+03 3.6220000E+03
FPR2      = 41100000 F0000000 1.000000E+00 1.0000000E+00
FPR4      = 00000000 00000000 0 0
FPR6      = 00000000 00000000 0 0
00000600 06 MODOUT - - -ADMDGW3 96067 V3R2.0

GENERAL PURPOSE REGISTERS
GPR0      = 007B4888 006E9450 00599C70 006D9BB8
GPR4      = 005993E0 00160A95 006E948B 0059A038
GPR8      = 001C5E85 006D9398 001C4E86 701C2E88
GPR12     = 001C3E87 006E9510 701C2EE4 801C96A8
FLOATING POINT REGISTERS
FPR0      = 46000120 00000000 2.880000E+02 2.8800000E+02
FPR2      = 43320000 00000000 8.000000E+02 8.0000000E+02
FPR4      = 00000000 00000000 0 0
FPR6      = C31207FE 00000000 -2.884995E+02 -2.8849950E+02

00000700 08 SUBOUT - - -ADMDGWI 96067 V3R2.0

GENERAL PURPOSE REGISTERS
GPR0      = 00000001 0016D9D8 005AA018 00000001
GPR4      = 0C0C0307 006E9748 006E9A30 0C0C0900
GPR8      = 00000014 006D9398 0016DBCE 7016BBD0
GPR12     = 0016CBCF 006F5028 6016BC56 8015A120
FLOATING POINT REGISTERS
FPR0      = 43120000 34000000 2.880000E+02 2.8800000E+02
FPR2      = 00000000 00000000 0 0
FPR4      = 00000000 00000000 0 0
FPR6      = 4311F000 00000000 2.870000E+02 2.8700000E+02
00000800 10 MODOUT - - - ADMERO0V 96067 V3R2.0
              CPNOUT - - - ESSFI ('080C0840'X) - FREE INSTANCE STG

```

```

GENERAL PURPOSE REGISTERS
GPR0   = 00667A60 00667A38 005993E0 00000001
GPR4   = 0C0C0307 0017B8E0 006F5318 005ACE08
GPR8   = 006E8000 006D9398 001C208E 401C0090
GPR12  = 001C108F 00667808 501C197C 800724B8
FLOATING POINT REGISTERS
FPR0   = 4311F000 00000000 2.870000E+02 2.870000E+02
FPR2   = 00000000 00000000 0 0
FPR4   = 407FFFFF 00000000 4.999999E-01 4.999999E-01
FPR6   = 4311F000 00000000 2.870000E+02 2.870000E+02
00000900 06 MODOUT - - -ADMLR01V 96067 V3R2.0
          CPNOUT - - -LCONV ('04000500'X) - CONVERSE
GENERAL PURPOSE REGISTERS
GPR0   = 005E7F98 006E995C 006D9688 006E98BB
GPR4   = 006D9BB8 006DAFA8 00000001 00000000
GPR8   = 000CBBA0 006D9398 000C9BA2 000CABA1
GPR12  = 000CB883 006E98D8 500C9FB6 800CE000
FLOATING POINT REGISTERS
FPR0   = 4E000000 0000011F 0 2.870000E+02
FPR2   = 00000000 00000000 0 0
FPR4   = 407FFFFF 00000000 4.999999E-01 4.999999E-01
FPR6   = 00000000 00000000 0 0

00001000 14 SUBOUT - - - - -ADMDGPA 96067 V3R2.0
GENERAL PURPOSE REGISTERS
GPR0   = 006D9BB8 00160A18 006E8000 006E9930
GPR4   = 00000000 00160A95 00000000 0C0C0009
GPR8   = 00599C70 006D9398 0015FA96 4015DA98
GPR12  = 0015EA97 006F5250 40160596 801685B0
FLOATING POINT REGISTERS
FPR0   = 00000000 00000000 0 0
FPR2   = 00000000 00000000 0 0
FPR4   = 00000000 00000000 0 0
FPR6   = 00000000 00000000 0 0
00001100 12 SUBOUT - - - - -ADMDGWI 96067 V3R2.0
GENERAL PURPOSE REGISTERS
GPR0   = 007B6174 00151014 0C240101 00000000
GPR4   = 000C46F8 006E97F8 00000000 000C46F8
GPR8   = 0C040002 006D9398 006D9BB8 401500B8
GPR12  = 001510B7 006E9930 70150BE4 8015A120
FLOATING POINT REGISTERS
FPR0   = 00000000 00000000 0 0
FPR2   = 00000000 00000000 0 0
FPR4   = 00000000 00000000 0 0
FPR6   = 00000000 00000000 0 0
00001200 07 MODIN - - - ADMEPR0V 96067 V3R2.0

```

```

GENERAL PURPOSE REGISTERS
GPR0   = 00000002 006E95F8 00000000 00000004
GPR4   = 000C46F8 006E8958 0C100000 006E9610
GPR8   = 000EA39C 006D9398 000734E1 400724E2
GPR12  = 00000020 006E9660 40072700 80074220
FLOATING POINT REGISTERS
FPR0   = 00000000 00000000                0                0
FPR2   = 00000000 00000000                0                0
FPR4   = 00000000 00000000                0                0
FPR6   = 00000000 00000000                0                0
00001300 04 MODOUT - -ADMER00V 96067 V3R2.0
                CPNOUT - -EPSDLG ('08040880'X) - GROUP DELETE MODULE (ADMLR01V)
GENERAL PURPOSE REGISTERS
GPR0   = 007B6174 006E8A60 000C46F8 006E8740
GPR4   = 006E8740 000C79D9 006BD038 006E8000
GPR8   = 006E8B01 006D9398 000C7868 400C5D98
GPR12  = 000C6D97 006E8958 400C6FBA 800724B8
FLOATING POINT REGISTERS
FPR0   = 00000000 00000000                0                0
FPR2   = 00000000 00000000                0                0
FPR4   = 00000000 00000000                0                0
FPR6   = 00000000 00000000                0                0

00001391 END OF GDDM DIAGNOSTIC TRACE 07 MAR 1996 8:49 AM

```

Example 5. Tracing calls to the 5080 Graphics System workstation

The following statement specifies a maximum of 200 bytes per record to be displayed in the trace output.

```
[label] ADMMDFT TRCESTR='5080IO(ON,200)'
```

The trace output for this statement is:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN HPV20                07 MAR 1996 11:31 AM
TRACE WORD = '00000000'X  BUILDID = 96067
```

```

*-----*
*
* Trace defaults and nickname table omitted for clarity
*
*-----*

```

```
DEFAULT TRCESTR='5080IO(ON,200)'
```

```

CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPOPPH ('0C000101'X)
PTRACE 1 CHAR 'PHIGSERR' D7C8C9C7 E2C5D9D9
PTRACE 2 FIXED 0
CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPEHND ('0C002C01'X)
PTRACE 1 FIXED 10,868,920
PTRACE OUTPUT PASS GPEHND ('0C002C01'X)
PTRACE 1 FIXED ---INPUT ONLY PARAMETER-----
CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPOPWS ('0C000301'X)
PTRACE 1 FIXED 1
PTRACE 2 CHAR 'IBM5080 ' C9C2D4F5 F0F8F040
PTRACE 3 CHAR 'GDDM5080' C7C4C4D4 F5F0F8F0

```

```

*-----*
*
* Trace records omitted for clarity
*
*-----*

CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPCLST ('0C001302'X)
ADMDGP5 CALLED WITH GDF CODE=
002C66D0 00000004 *... *
ADMDGP5 CALLED WITH GDF CODE=
002C66D0 0000003E *... *

ADMDGP5 CALLED WITH GDF CODE=
002C66D0 00000033 *... *
ADMDGP5 CALLED WITH GDF CODE=
002C66D0 000000C3 *...C *

CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPOPST ('0C001301'X)
PTRACE 1 FIXED 2,898,616
CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPAN2 ('0C00060B'X)
PTRACE 1 FLOAT 1,934.437 2,072.062
PTRACE 2 DIM 4
PTRACE 3 CHAR 'HHHH' C8C8C8C8
CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPCLST ('0C001302'X)

ADMDGP5 CALLED WITH GDF CODE=
002C66D0 000000C1 *...A *

ADMDGP5 CALLED WITH GDF CODE=
002C66D0 000000C1 *...A *

ADMDGP5 CALLED WITH GDF CODE=
002C66D0 000000C1 *...A *

ADMDGP5 CALLED WITH GDF CODE=
002C66D0 000000C1 *...A *

ADMDGP5 CALLED WITH GDF CODE=
002C66D0 00000071 *... *

CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPOPST ('0C001301'X)
PTRACE 1 FIXED 2,898,616
CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPDPL2 ('0C000611'X)
PTRACE 1 DIM 5
PTRACE 2 FIXED 2
PTRACE 3 FLOAT 0 0 3,840 0
PTRACE 3 FLOAT 3,840 4,095.999 0 4,095.999
PTRACE 3 FLOAT 0 0
PTRACE 4 FIXED 2 2 2 2
PTRACE 4 FIXED 0

CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS GPCLST ('0C001302'X)

*-----*
*
* Trace records omitted for clarity
*
*-----*

000004693 END OF GDDM DIAGNOSTIC TRACE 05 SEP 1988 11:31 AM

```

Example 6. Tracing in multiple instances of GDDM

The number in columns 2 through 4 shows in which instance of GDDM the event that caused the trace occurred.

```

1001  DIAGNOSTIC TRACE - ORIGIN JHERROD           07 MAR 1996 10:32 AM
001
001  TRACE WORD = '07F80003'X   BUILDID = 96067
001
001  00000001 01 CPNIN  FSINIT ('0C000001'X) - INITIALIZATION
001              CPNOUT FSINIT ('0C000001'X) - INITIALIZATION
001
001  00000025 01 CPNIN  DSOPEN ('0C000200'X) - OPEN DEVICE
001              CPNOUT DSOPEN ('0C000200'X) - OPEN DEVICE
001
001  00000237 01 CPNIN  DSUSE  ('0C000202'X) - DEVICE USAGE
001              CPNOUT DSUSE  ('0C000202'X) - DEVICE USAGE
001
001  00000241 01 CPNIN  FSQURY ('0C040501'X) - EXTENDED QUERY DEVICE
001              CPNOUT FSQURY ('0C040501'X) - EXTENDED QUERY DEVICE
001
001  00000259 01 CPNIN  WSCRT  ('0C2C0000'X) - CREATE OPERATOR WINDOW
001              CPNOUT WSCRT  ('0C2C0000'X) - CREATE OPERATOR WINDOW
001
001  00000267 01 CPNIN  DSOPEN ('0C000200'X) - OPEN DEVICE
001              CPNOUT DSOPEN ('0C000200'X) - OPEN DEVICE
001
001  00000381 01 CPNIN  DSUSE  ('0C000202'X) - DEVICE USAGE
001              CPNOUT DSUSE  ('0C000202'X) - DEVICE USAGE
001
001  00000385 01 CPNIN  ASDFLD ('0C080700'X) - DEFINE ALPHA FIELD
001              CPNOUT ASDFLD ('0C080700'X) - DEFINE ALPHA FIELD
001
001  00000475 01 CPNIN  ASFCOL ('0C080502'X) - SET FIELD COLOR
001              CPNOUT ASFCOL ('0C080502'X) - SET FIELD COLOR
*-----*
*
*  Trace records omitted for clarity
*
*-----*
001
001  00002633 01 CPNIN  WSCRT  ('0C2C0000'X) - CREATE OPERATOR WINDOW
001              CPNOUT WSCRT  ('0C2C0000'X) - CREATE OPERATOR WINDOW
001
1002  DIAGNOSTIC TRACE - ORIGIN JHERROD           07 MAR 1996 10:32 AM
002
002  TRACE WORD = '07F80003'X   BUILDID = 96067
002
002  00000001 01 CPNIN  SPINIT ('00050000'X) - SPI SPECIAL INIT
002              CPNOUT SPINIT ('00050000'X) - SPI SPECIAL INIT
002
002  00000039 01 CPNIN  DSOPEN ('0C000200'X) - OPEN DEVICE
002              CPNOUT DSOPEN ('0C000200'X) - OPEN DEVICE
002
002  00000149 01 CPNIN  DSUSE  ('0C000202'X) - DEVICE USAGE
002              CPNOUT DSUSE  ('0C000202'X) - DEVICE USAGE
002
002  00000161 01 CPNIN  CSCCRT ('14040000'X) - CREATE CHART
002              CPNOUT CSCCRT ('14040000'X) - CREATE CHART
002
001  00002671 01 CPNIN  WSIO   ('0C100008'X) - WINDOWED DEVICE I/O
001              CPNOUT WSIO   ('0C100008'X) - WINDOWED DEVICE I/O
001
002

```

```

002 00003897 01 CPNIN  FSTERM ('0C000000'X) - TERMINATION
002                CPNOUT FSTERM ('0C000000'X) - TERMINATION
002
002 00004172 END OF GDDM DIAGNOSTIC TRACE  07 MAR 1996 10:33 AM
001
001 00002867 01 CPNIN  WSDDEL ('0C2C0100'X) - DELETE OPERATOR WINDOW
001                CPNOUT WSDDEL ('0C2C0100'X) - DELETE OPERATOR WINDOW
001
001 00002879 01 CPNIN  WSSWP  ('0C2C0900'X) - SET WINDOW PRIORITIES
001                CPNOUT WSSWP  ('0C2C0900'X) - SET WINDOW PRIORITIES
001
001 00002885 01 CPNIN  ASCPUT ('0C080603'X) - SET CHARACTER CODES
001                CPNOUT ASCPUT ('0C080603'X) - SET CHARACTER CODES
001
001 00002895 01 CPNIN  WSIO   ('0C100008'X) - WINDOWED DEVICE I/O
001                CPNOUT WSIO   ('0C100008'X) - WINDOWED DEVICE I/O
001
001 00003301 01 CPNIN  ASREAD ('0C100000'X) - READ
001                CPNOUT ASREAD ('0C100000'X) - READ
001
001 00003319 01 CPNIN  FSTERM ('0C000000'X) - TERMINATION
001                CPNOUT FSTERM ('0C000000'X) - TERMINATION
001
001 00003892 END OF GDDM DIAGNOSTIC TRACE  07 MAR 1996 10:33 AM

```

Example 7. Trace output containing CDPDS input

If you specify CDPDS in your TRCESTR statements, the trace output contains fields in the following form:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN BMASTERS          07 MAR 1996  3:20 PM
```

```
TRACE WORD = '00000000'X  BUILDID = 96067
```

```

*-----*
*
* Trace defaults and nickname table omitted for clarity
*
*-----*

CDPDS structured field ...
00362FF0  0058D3EE  EE000000  C3C4D7C4  E2409799  *..L.....CDPDS pr*
00363000  9684A483  85844082  A8400000  00000000  *oduced by .....*
00363010  00000000  40404040  40404040  D6D540F0  *....      ON 0*
00363020  F961F1F1  61F8F640  40F1F57A  F2F44040  *9/11/86  15:24 *
00363030  C2E840C2  D4C1E2E3  C5D9E240  E5D4F3F8  *BY BMASTERS VM38*
00363040  F1F24040  40404040  *12          *

CDPDS structured field ...
00362FF0  0062D3A8  A8000000  E2E2E240  40404040  *..Lyy...SSS   *
00363000  00000601  01510100  08210200  80000000  *.....*
00363010  08210600  80000000  08210300  80004000  *..... .*
00363020  05234100  0005230A  00000523  06000005  *.....*
00363030  23070000  05230100  00052308  00000523  *.....*
00363040  02800005  230F0000  05230300  00051801  *.....*
00363050  0800          *..          *

CDPDS structured field ...
00362FF0  0010D3A8  AF000000  F1404040  40404040  *..Ly....1     *

CDPDS structured field ...
00362FF0  0010D3A8  C9000000  F0404040  40404040  *..LyI...0     *

CDPDS structured field ...
00362FF0  0033D3AB  8A000000  002B0C02  8500E3F1  *..L.....e.Tl*
00363000  C4F0C2C1  E2C50C02  8600C3F1  E2F0D7D9  *D0BASE..f.C1S0PR*
00363010  F1F20424  05010D1F  0505005A  003C0000  *12.....!.....*

```



```

00363020    090003                *...                *
CDPDS structured field ...
00362FF0    0014D3A6 AF000000  00003840  3840002F  *..Lw.....  ..*
00363000    D0003DE0                *}..\                *
CDPDS structured field ...
00362FF0    0012D3B1 AF000000  01090000  00000000  *..L.....*
00363000    0000                *..                *
CDPDS structured field ...
00362FF0    001CD3A6 6B000000  03430108  4B000038  *..Lw,.....*
00363000    40384009 4C02002F  D0003DE0                * . .<...}..\  *
CDPDS structured field ...
00362FF0    0020D3AC 6B000000  01170000  00000000  *..L,.....*
00363000    00002D00 00000000  00000000  002D0001  *.....*
CDPDS structured field ...
00362FF0    0016D3A6 9B000000  00003840  3840002F  *..Lw.....  ..*
00363000    D0003DE0 0000                *}..\..                *
CDPDS structured field ...
00362FF0    0010D3A9 C9000000  F0404040  40404040  *..LzI...0      *
CDPDS structured field ...
00362FF0    0010D3A8 9B000000  F1404040  40404040  *..Ly....1      *
CDPDS structured field ...
00362FF0    0046D3EE 9B000000  2BD304D3  069004C7  *..L.....L.L...G*
00363000    061804C5 007803F1  0106DBE3  8889A204  *...E...1...This.*
00363010    C9007804 DB89A204  C9007803  DB8104C9  *I....is.I....a.I*
00363020    007808DB A2819497  938504C9  007806DB  *....sample.I....*
00363030    86899385 02F8                *file.8          *
CDPDS structured field ...
00362FF0    0010D3A9 9B000000  F1404040  40404040  *..Lz....1      *
CDPDS structured field ...
00362FF0    0010D3A9 AF000000  F1404040  40404040  *..Lz....1      *
CDPDS structured field ...
00362FF0    0010D3A9 A8000000  E2E2E240  40404040  *..Lzy...SSS    *

00001075 END OF GDDM DIAGNOSTIC TRACE  07 MAR 1996  3:20 PM

```

Notes:

1. If the field header is invalid, an entry appears in the form:

CDPDS field header invalid ...

Only the first eight bytes of the structured field are included in the output.

2. If there are less than eight bytes in a record, an entry appears in the form:

CDPDS short record ...

The data in the short record is included in the output.

Example 8. Trace output containing HRIG output

If you specify HRIG in your TRCESTR statements, the trace output contains fields in the following form:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN BMASTERS          07 MAR 1996  2:36 PM
```

```
TRACE WORD = '00000000'X  BUILDID = 96067
```

```

*-----*
*
* Trace defaults and nickname table omitted for clarity
*
*-----*

```

Begin Document - First Page

```

TRANSMIT FIRST
XMIT-----
00363FE8          5A0058D3 EEEE0000 *      !..L....*
00363FF0    00D7D9C9 D5E340C7 C5D5C5D9 C1E3C5C4 *.PRINT GENERATED*
00364000    40D6D540 F1F140E2 C5D740F1 F9F8F640 * ON 07 MAR 1996 *
00364010    40F37AF3 F040D7D4 40C2E840 C2D4C1E2 * 3:30 PM BY BMAS*
00364020    E3C5D9E2 40404040 40404040 40404040 *TERS *
00364030    40404040 40404040 40404040 40404040 * *
00364040    40 * *
TRANSMIT MIDDLE
XMIT-----
00363FE8          5A0010D3 A8A80000 *      !..Lyy..*
00363FF0    00C4D6C3 40404040 40 *.DOC *
TRANSMIT MIDDLE
XMIT-----
00363FE8          5A0010D3 A8AF0000 *      !..Ly...*
00363FF0    00F14040 40404040 40 *.1 *
TRANSMIT MIDDLE
XMIT-----
00363FE8          5A0010D3 A8C90000 *      !..LyI..*
00363FF0    00C1C5C7 40404040 40 *.AEG *
TRANSMIT MIDDLE
XMIT-----
00363FE8          5A0028D3 B18A0000 *      !..L....*
00363FF0    001C0000 00010000 00FFFFFF FFFFFFFF *.....*
00364000    FFE3F1C4 F0C2C1E2 C5C3F1E2 F0D7D9F1 *.T1D0BASEC1S0PR1*
00364010    F2 *2 *
TRANSMIT MIDDLE
XMIT-----
00363FE8          5A0012D3 EE9B0000 *      !..L....*
00363FF0    002BD304 D1FFFF04 C0FFFF *..L.J...{.. *
TRANSMIT MIDDLE
XMIT-----
00363FE8          5A0046D3 EE9B0000 *      !..L....*
00363FF0    002BD304 D3011804 C7010404 C5001403 *..L.L...G...E...*
00364000    F10106DB E38889A2 04C90014 04DB89A2 *1...This.I....is*
00364010    04C90014 03DB8104 C9001408 DBA28194 *.I....a.I....sam*
00364020    97938504 C9001406 DB868993 8502F8 *ple.I....file.8 *
TRANSMIT MIDDLE
XMIT-----
00363FE8          5A0010D3 A99B0000 *      !..Lz...*
00363FF0    00E3E7E3 40404040 40 *.TXT *
End Document
TRANSMIT LAST
XMIT-----
00363FE8          5A0010D3 A9A80000 *      !..Lzy..*
00363FF0    00C4D6C3 40404040 40 *.DOC *

00001075 END OF GDDM DIAGNOSTIC TRACE 07 MAR 1996 2:36 PM

```

Example 9. Trace output containing PostScript output

To trace the creation of PostScript output specify the POSTSCR function on the TRCESTR external default. This includes PostScript debug information in the trace output. To include the PostScript output itself in the trace in EBCDIC form, specify the FULLIO function on the TRCESTR external default.

```

1GDDM DIAGNOSTIC TRACE - ORIGIN WILLMSB          29 APR 1996  5:33 PM

TRACE WORD = '00000000'X   BUILDID = 96100
*-----*
*
*   Trace external-default and nickname tables omitted for clarity
*
*-----*

DEFAULT TRCESTR='POSTSCR '

color table create
color table created ok at 0081A008
Symbol Set to Font table create
Symbol Set to Font table created ok at 0081B008
Pres text to Font table create
Pres text to Font table created ok at 00817110
Comment Order
GSQFLD: ROW=          1 COL=          1 WID=          85 DEP=          94
PSWIDTH=          595 PSDEPTH=          841 LCLPGW=          85 LCLPGD=          94
DEVR_AlphaW=          85 DEVR_AlphaD=          94
Xlower= -2465,Xupper= 2464 Ylower= -1726,Yupper= 1725
ScaleX= 1.2071410E-01,ScaleY= 2.4369750E-01
SCALEX used
Begin Segment Order
Line width 1 found in order
gdf Other attrs 3E not processed
Segment viewing mask =00111100
SVL XL= -2465 XR= 2462 YB= -3478 YT= 3475
Error pushed order found!
gdf tag attr 43 not processed
pattern attribute 16 stored
color 7 found in order
Mix attribute RPGD_MIX=00
Begin Area Order
Line type 0 found in order
Current point before primitive pos X=          0 Y=          0
Current point after primitive pos X=          -989 Y=          -108
BBOX set PBBX before transform =          -989 PBBY=          -108
Line order found
BBOX set PBBX before transform =          -989 PBBY=          99
line to X=          -989 Y=          99
BBOX set PBBX before transform =          670 PBBY=          99
line to X=          670 Y=          99
BBOX set PBBX before transform =          670 PBBY=          305
line to X=          670 Y=          305
BBOX set PBBX before transform =          980 PBBY=          -4
line to X=          980 Y=          -4
BBOX set PBBX before transform =          670 PBBY=          -314
line to X=          670 Y=          -314
BBOX set PBBX before transform =          670 PBBY=          -108
line to X=          670 Y=          -108
BBOX set PBBX before transform =          -989 PBBY=          -108
line to X=          -989 Y=          -108
Current point after primitive X=          -989 Y=          -108
within area - End Area Order
pattern 16 not found
BBOX set PBBX before transform =          -989 PBBY=          -108
Line order found
BBOX set PBBX before transform =          -989 PBBY=          99
line to X=          -989 Y=          99
BBOX set PBBX before transform =          670 PBBY=          99
line to X=          670 Y=          99
BBOX set PBBX before transform =          670 PBBY=          305
line to X=          670 Y=          305
BBOX set PBBX before transform =          980 PBBY=          -4
line to X=          980 Y=          -4
BBOX set PBBX before transform =          670 PBBY=          -314
line to X=          670 Y=          -314
BBOX set PBBX before transform =          670 PBBY=          -108
line to X=          670 Y=          -108
BBOX set PBBX before transform =          -989 PBBY=          -108
line to X=          -989 Y=          -108
Current point after primitive X=          -989 Y=          -108
Segment End Order
End of segment reset RPS CS=00
End of seg RPGD X=          -989 Y=          -108, RPSS X=          32767 Y=          32767
RPGD BBOX LLX=          -989, LLY=          -314, URX=          980, URY=          305
RPS BBOX LLX=          0, LLY=          0, URX=          0, URY=          0
BBOX SET LLX=          178, LLY=          382, URX=          416, URY=          458
New RPS BBOX LLX=          178, LLY=          382, URX=          416, URY=          458
9 00011099 END OF GDDM DIAGNOSTIC TRACE  29 APR 1996  5:34 PM

```

Example 10. Tracing an imported CGM

The statement for example 10 is:

```
DEFAULT TRCESTR='CGMREP(ON)'
```

The trace output is:

```
GDDM DIAGNOSTIC TRACE - ORIGIN BLAND          11 APR 1996 11:16 AM

TRACE WORD = '00000000'X   BUILDID = 96077
*****
*
* Trace defaults and nickname table omitted from this example
* for clarity
*
*****
DEFAULT TRCESTR='CGMREP(ON)'
```

Using CGM conversion profile: ADM ADMCGM *

```
*****
*
* The CGM Conversion Profile omitted from this example for clarity.
* The complete profile used would be shown, including comment lines.
*
*****
```

Completed CGM conversion profile: ADM ADMCGM *

```
*****
*
* The next "header" line is placed in the trace before every 10 CGM
* orders that are found in the imported CGM.
*
* Order # - the sequential count of CGM elements (commands) found
* Class   - the CGM element class.
* Element - the CGM element ID.
* Parmct  - the number of bytes in the CGM order.
* Record  - the record number in which the CGM element started.
* Cursize - the length of the records in the CGM. Hex 190 = decimal 400
* Index   - the offset in the Record where the CGM element starts
*
* Refer to the ISO/IEC 8632-3 standard, and GDDM Base Application
* Programming Reference, for more details.
*****
CGMREP: Order # Class Element Parmct Record Cursize Index
CGMREP: 00000001 00000000 00000001 0000001A 00000001 00000190 00000002
CGMREP: 00000002 00000001 00000001 00000002 00000001 00000190 0000001E
*****
* Parameter details follow the CGM elements, where appropriate.
*****
CGMREP: II 00000001
CGMREP: 00000003 00000001 00000002 00000028 00000001 00000190 00000024
CGMREP: 00000004 00000001 00000003 00000002 00000001 00000190 0000004E
CGMREP: INT 00000000
CGMREP: 00000005 00000001 00000004 00000002 00000001 00000190 00000052
CGMREP: II 00000010
CGMREP: 00000006 00000001 00000005 00000006 00000001 00000190 00000056
CGMREP: INT 00000001
CGMREP: II 00000010
CGMREP: II 00000010
CGMREP: 00000007 00000001 00000006 00000002 00000001 00000190 0000005E
```

```

| CGMREP: II 00000010
| CGMREP: 00000008 00000001 00000007 00000002 00000001 00000190 00000062
| CGMREP: II 00000010
| CGMREP: 00000009 00000001 00000008 00000002 00000001 00000190 00000066
| CGMREP: II 00000010
| CGMREP: 0000000A 00000001 00000009 00000002 00000001 00000190 0000006A
| CGMREP: ICI 0000009A
| CGMREP: Order # Class Element Parmct Record Cursize Index
| CGMREP: 0000000B 00000001 0000000A 0000000C 00000001 00000190 0000006E
| CGMREP: ICD 00000000
| CGMREP: ICD 00000000
| CGMREP: ICD 00000000
| CGMREP: ICD 000000FF
| CGMREP: ICD 000000FF
| CGMREP: ICD 000000FF
| CGMREP: 0000000C 00000001 0000000B 000000DE 00000001 00000190 0000007E
| CGMREP: II 00000037
| CGMREP: IIX 00000000
| CGMREP: IIX 00000000
| *****
| * Trace records omitted for clarity *
| *****
| CGMREP: IIX 00000007
| CGMREP: IIX 00000002
| CGMREP: 0000000D 00000001 0000000D 0000006C 00000001 00000190 00000160
| *****
| * The font names in the CGM Font List are shown, and are useful when *
| * tailoring the CGM_FONT_NAME and GDF_FONT_NAME entries in the *
| * CGM Conversion Profile. *
| *****
| CGMREP: Item 1 in CGM Font List is Helv
| CGMREP: Item 2 in CGM Font List is Tms Rmn
| CGMREP: Item 3 in CGM Font List is Courier
| CGMREP: Item 4 in CGM Font List is Helvetica
| CGMREP: Item 5 in CGM Font List is Helvetica Narrow
| CGMREP: Item 6 in CGM Font List is LotusLineDraw
| CGMREP: Item 7 in CGM Font List is LotusWFont
| CGMREP: Item 8 in CGM Font List is Symbol
| CGMREP: Item 9 in CGM Font List is Symbol Set
| CGMREP: Item 10 in CGM Font List is Times
| CGMREP: Item 11 in CGM Font List is Times New
| CGMREP: 0000000E 00000007 00000002 00000004 00000002 00000190 0000003E
| CGMREP: 0000000F 00000000 00000003 0000000A 00000002 00000190 00000044
| CGMREP: 00000010 00000002 00000001 00000006 00000002 00000190 00000050
| CGMREP: INT 00000000
| CGMREP: 00000011 00000002 00000002 00000002 00000002 00000190 00000058
| CGMREP: INT 00000000
| CGMREP: 00000012 00000002 00000003 00000002 00000002 00000190 0000005C
| CGMREP: INT 00000000
| CGMREP: 00000013 00000002 00000004 00000002 00000002 00000190 00000060
| CGMREP: INT 00000000
| CGMREP: 00000014 00000002 00000005 00000002 00000002 00000190 00000064
| CGMREP: INT 00000000
| CGMREP: Order # Class Element Parmct Record Cursize Index
| CGMREP: 00000015 00000002 00000006 00000008 00000002 00000190 00000068
| CGMREP: RP2 0 0
| CGMREP: RP2 32,767 23,135
| CGMREP: 00000016 00000002 00000007 00000006 00000002 00000190 00000072
| CGMREP: COL FFFFFFFE 0B453724
| CGMREP: 00000017 00000000 00000004 00000000 00000002 00000190 0000007A
| CGMREP: 00000018 00000005 00000022 0000039E 00000002 00000190 0000007E
| CGMREP: ICI 00000000
| CGMREP: MAPI00000085
| CGMREP: MAPIFFFFFFFE

```

tracing

```
| CGMREP: MAPIFFFFFFF
| *****
| * Trace records omitted for clarity *
| *****
| CGMREP: MAPI0000009A
| CGMREP: MAPI00000090
| CGMREP: 00000019 00000003 00000001 00000002 00000004 00000190 000000FE
| CGMREP: II 00000010
| *****
| * Trace records omitted for clarity *
| *****
| CGMREP: 0000002F 00000005 00000010 00000008 00000004 00000190 00000186
| CGMREP: RP2 0 1,087
| CGMREP: RP2 1,087 0
| CGMREP: 00000030 00000004 00000004 00000010 00000004 00000190 00000190
| CGMREP: RP2 16,424 10,475
| CGMREP: INT 00000001
| CGMREP: 00000031 00000000 00000005 00000000 00000005 00000190 00000012
| CGMREP: 00000032 00000000 00000002 00000000 00000005 00000190 00000014
|
| 00010501 END OF GDDM DIAGNOSTIC TRACE 11 APR 1996 11:16 AM
```

Example 11. Tracing an exported CGM

The statement for example 11 is:

```
DEFAULT TRCESTR='CGMREP(ON) '
```

The trace output is:

```
| GDDM DIAGNOSTIC TRACE - ORIGIN BLAND 11 APR 1996 2:53 PM
|
| TRACE WORD = '00000000'X BUILDID = 96077
| *****
| * *
| * Trace defaults and nickname table omitted from this example *
| * for clarity *
| * *
| *****
| DEFAULT TRCESTR='CGMREP(ON) '
| Using CGM conversion profile: ADM ADMCGM *
| *****
| * *
| * The CGM Conversion Profile omitted from this example for clarity. *
| * The complete profile used would be shown, including comment lines. *
| * *
| *****
| Completed CGM conversion profile: ADM ADMCGM *
| *****
| * The Destination CGM file is specified: *
| *****
| CGMREP: TEMP CGM A
| *****
| * The building up of the CGM element is shown in hexadecimal, (using *
| * ASCII codes, not EBCDIC). This is followed by the completed CGM *
| * element in human-readable form, starting with the CGM element Class *
| * and element ID. Refer to the ISO/IEC 8632-3 standard for details. *
| *****
| CGMREP: Outputting: 003F0031
```

```

| CGMREP: Outputting: 30
| CGMREP: Outputting: 43726561 74656420 62792041 444D5547 43205633 52322E30 20667
| CGMREP: (0,1) Begin Metafile:"Created by ADMUGC V3R2.0 from GDF file SAMPLE  "
| CGMREP: Outputting: 10220001
| CGMREP: (1,1) Metafile Version=0001
| CGMREP: Outputting: 105F003A
| CGMREP: Outputting: 39
| CGMREP: Outputting: 41444D43 474D2047 44444D20 20563352 322E3020 39362031 30322
| CGMREP: (1,2) Metafile Description:"ADMCGM GDDM V3R2.0 96 102 14:53:31
| CGMREP: Outputting: 10620000
| CGMREP: (1,3) VDC Type=0000
| CGMREP: Outputting: 10E20010
| CGMREP: (1,7) Color Prec.=0010
| CGMREP: Outputting: 11020010
| CGMREP: (1,8) Color Index Prec.=0010
| CGMREP: Outputting: 112200FF
| CGMREP: (1,9) Maximum Color Index=00FF
| CGMREP: Outputting: 114C
| CGMREP: Outputting: 00000000 000000FF 00FF00FF
| CGMREP: (1,10) RGB Color Value Extents=00000000 000000FF 00FF00FF
| CGMREP: Outputting: 11660001
| CGMREP: Outputting: FFFF0001
| CGMREP: (1,11) Metafile Element List=FFFF0001
| CGMREP: Outputting: 11BF00BA
| CGMREP: Outputting: 0B54494D 45535F52 4F4D414E 0C54494D 45535F49 54414C49 430A5
| CGMREP: (1,13) Font List=" TIMES_ROMAN TIMES_ITALIC TIMES_BOLD TIMES_BO..."
| CGMREP: Outputting: 0069
| CGMREP: Outputting: 08
| CGMREP: Outputting: 50494354 55524531 00
| CGMREP: (0,3) Begin Picture:"PICTURE1" 000A
| CGMREP: Outputting: 20620000
| CGMREP: (2,3) Line Width Spec. Mode=0000
| CGMREP: Outputting: 20820000
| CGMREP: (2,4) Marker Size Spec. Mode=0000
| CGMREP: Outputting: 20A20000
| CGMREP: (2,5) Edge Width Spec. Mode=0000
| *****
| * The attribute conversions resulting from the conversion profile and *
| * standard defaults is shown in the trace: *
| *****
| CGMREP: ADMDGM0: Attribute Conversion Table Built, trace subset 0:160000003C
| CGMREP: GDF Index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
| CGMREP: CGM_COLOR:00030004 00020006 00030007 00050001 00000009 000A000B 000C000
| CGMREP: LINE_TYPE:00010003 00000004 00000000 00050001 0000
| CGMREP: LINE_MODE:00010001 00000001 00000000 00010001 0000
| CGMREP: Int_STYLE:00010003 00030003 00030003 00030003 00030003 00030003 0003000
| CGMREP: CGM_HATCH:00000001 00010001 00010005 00060001 00010002 00010003 0001000
| CGMREP: FILL_MODE:00010000 00000000 00000001 00010000 00000001 00010001 0000000
| CGMREP: MARK_TYPE:00050005 00020001 00010001 00030001 00010001 0004
| CGMREP: MARK_MODE:00010001 00010000 00000000 00010000 00000001 0001
| *****
| * The GDF orders of the source picture are shown, and their effect on *
| * the target CGM: *
| *****
| CGMREP: GDF Buffer Address=0098D000
| CGMREP: GDF length=000001C6
| CGMREP: GDF Order=01120004 00000000 42640000 00000000 42640003
| CGMREP: GDF Order=0202017E

```

tracing

```
| CGMREP: GDF Order=02150140 020071C1 C4D4E4E4 C8404040 40404040 404040
| CGMREP: GDF Order=0202017F
| CGMREP: GDF Order=0202027E
| CGMREP: GDF Order=0203020E 80
| CGMREP: GDF Order=02150232 12003C00 00000042 64000000 00000042 640003
| CGMREP: XLO,XHI,YLO,YHI=0 100 0 100.0001
| CGMREP: X Scale,Y Scale=0 0
| CGMREP: X Offset,Y Offset=0 0
| CGMREP: CHAR W & H=1.11111 3.125
| CGMREP: GDF Order=02150227 12003C00 00000042 64000000 00000042 640003
| CGMREP: XLO,XHI,YLO,YHI=0 100 0 100.0001
| CGMREP: X Scale,Y Scale=0 0
| CGMREP: X Offset,Y Offset=0 0
| CGMREP: CHAR W & H=1.11111 3.125
| CGMREP: GDF Order=020B0220 08406B86 E0409735 B0
| CGMREP: GDF Order=020B8001 08000000 00000000 00
| CGMREP: GDF Order=0202027F
| CGMREP: XLO,XHI,YLO,YHI=0 100 0 100.0001
| CGMREP: X Scale,Y Scale=327.67 233.0096
| CGMREP: X Offset,Y Offset=0 0
| CGMREP: CHAR W & H=1.11111 3.125
| CGMREP: GDF Order=700C0000 00005410 00000000 0000
| CGMREP: GDF Order=3E00
| CGMREP: GDF Order=0900
| CGMREP: GDF Order=66020007
| CGMREP: GDF Order=6880
| CGMREP: Outputting: 20C8
| CGMREP: Outputting: 00000000 7FFF5B05
| CGMREP: (2,6) VDC Extent=00000000 7FFF5B05
| CGMREP: Outputting: 20E6
| CGMREP: Outputting: 00FF00FF 00FF
| CGMREP: (2,7) Background Color=00FF00FF 00FF
| CGMREP: Outputting: 0080
| CGMREP: (0,4) Begin Picture Body 0080
| CGMREP: Outputting: 545F
| CGMREP: Outputting: 05FC
| CGMREP: Outputting: 00010000 000000FF 00FF0000 000000FF 000000FF 000000FF 000000
| CGMREP: (5,34) Color Table 00010000 000000FF 00FF0000 000000FF 000000FF 000000FF
| CGMREP: Init CGM Area Fill, NPAT=0000
| CGMREP: Fill Color=FFFE
| CGMREP: GDF Order=C1280000 0000405B 2A8E0000 00004263 D6CF4264 00004263 D6CF42
| CGMREP: Init Edge=0001
| CGMREP: Draw Edge=0002
| CGMREP: Draw Edge=0003
| CGMREP: Draw Edge=0004
| CGMREP: Draw Edge=0005
| CGMREP: GDF Order=6000
| CGMREP: Fill Area, NPAT=0000
| CGMREP: Fill color=FFFE
| CGMREP: Outputting: 52E2
| CGMREP: Outputting: 0007
| CGMREP: (5,23) Fill Color=0007
| CGMREP: Outputting: 52C2
| CGMREP: Outputting: 0001
| CGMREP: (5,22) Fill Style=0001
| CGMREP: Outputting: 40F0
| CGMREP: Outputting: 0000
| CGMREP: Outputting: 0053
```



```

| CGMREP: Outputting: 0000
| CGMREP: Outputting: 5ADF
| CGMREP: Outputting: 7FFF
| CGMREP: Outputting: 5ADF
| CGMREP: Outputting: 7FFF
| CGMREP: Outputting: 0053
| CGMREP:   Polygon Points=00000004
| CGMREP:   End Area Fill, NPAT=0000
| CGMREP:   Fill color=FFFE
| CGMREP: GDF Order=7100
| CGMREP: GDF Order=700C0000 00017418 00000000 0000
| CGMREP: GDF Order=040C8000 04080000 0000405B 2A8E
| CGMREP: GDF Order=3E00
| CGMREP: GDF Order=7A00
| CGMREP: GDF Order=23040000 0001
| CGMREP: GDF Order=4D05
| CGMREP: GDF Order=6712003C 00000000 42640000 405B2A8E 4263D6CF
| CGMREP: Outputting: 30A8
| CGMREP: Outputting: 00000053 7FFF5ADF
| CGMREP: (3,5) Clip Rectangle=00000053 7FFF5ADF
| CGMREP: GDF Order=7871
| CGMREP: GDF Order=7903
| CGMREP: GDF Order=66020008
| CGMREP: GDF Order=03084111 C71D4131 FFFD
| CGMREP: GDF Order=74084110 00000000 0000
| CGMREP: GDF Order=75083B65 791A4110 0000
| CGMREP: GDF Order=50020305
| CGMREP: GDF Order=0308414D A39A416D 64FF
| CGMREP: GDF Order=74084110 00000000 0000
| CGMREP: GDF Order=75083B65 791C4110 0000
| CGMREP: GDF Order=C30D4231 E69C423A 559DE389 A39385
| CGMREP:   FONT_CURR=0000
| CGMREP: Outputting: 5142
| CGMREP: Outputting: 0005
| CGMREP: (5,10) FONT INDEX=0005
| CGMREP: Outputting: 5162
| CGMREP: Outputting: 0002
| CGMREP: (5,11) Text Precision=0002
| CGMREP: Outputting: 51C2
| CGMREP: Outputting: 0008
| CGMREP: (5,14) Text Color=0008
| CGMREP: Outputting: 51E2
| CGMREP: Outputting: 0639
| CGMREP: (5,15) Char Height=0639
| CGMREP: Outputting: 5184
| CGMREP: Outputting: 0000FF80
| CGMREP: (5,12) Character Expansion=0000FF80
| CGMREP: Outputting: 5208
| CGMREP: Outputting: 00000639 06360000
| CGMREP: (5,16) Text Orientation=00000639 06360000
| CGMREP: Outputting: 524C
| CGMREP: Outputting: 00020004 00000000 00000000
| CGMREP: (5,18) Text Alignment=00020004 00000000 00000000
| CGMREP: Outputting: 408C
| CGMREP: Outputting: 3FDF
| CGMREP: Outputting: 3518
| CGMREP:   Text Coord:3FDF3518
| CGMREP: Outputting: 0001

```

tracing

```
| CGMREP: Outputting: 05
| CGMREP: Title
| CGMREP: Outputting: 5469746C 65
| CGMREP: (4,4) Output Text String, Nchr=00000005
| CGMREP: GDF Order=0308414D A39A416D 64FF
| CGMREP: GDF Order=74084110 00000000 0000
| CGMREP: GDF Order=75083B65 791C4110 0000
| CGMREP: GDF Order=50020303
| CGMREP: GDF Order=03084135 13EB414A C98C
| CGMREP: GDF Order=74084110 00000000 0000
| CGMREP: GDF Order=75083B65 791C4110 0000
| CGMREP: GDF Order=C3104232 1FA4422E 499DE2A4 82A389A3 9385
| CGMREP: FONT_CURR=0000
| CGMREP: Outputting: 51E2
| CGMREP: Outputting: 0441
| CGMREP: (5,15) Char Height=0441
| CGMREP: Outputting: 5208
| CGMREP: Outputting: 00000441 043F0000
| CGMREP: (5,16) Text Orientation=00000441 043F0000
| CGMREP: Outputting: 524C
| CGMREP: Outputting: 00020002 00000000 00000000
| CGMREP: (5,18) Text Alignment=00020002 00000000 00000000
| CGMREP: Outputting: 408F
| CGMREP: Outputting: 4028
| CGMREP: Outputting: 2A21
| CGMREP: Text Coord:40282A21
| CGMREP: Outputting: 0001
| CGMREP: Outputting: 08
| CGMREP: Subtitle
| CGMREP: Outputting: 53756274 69746C65 00
| CGMREP: (4,4) Output Text String, Nchr=00000009
| CGMREP: GDF Order=C1084237 923A4229 9D04
| CGMREP: GDF Order=7100
| CGMREP: Outputting: 30A8
| CGMREP: Outputting: 00000000 7FFF5B05
| CGMREP: (3,5) Clip Rectangle=00000000 7FFF5B05
| CGMREP: GDF Buffer Address=0098D000
| CGMREP: GDF length=00000000
| CGMREP: GDF Buffers processed=00000001
| CGMREP: ADMDGMD: End of Picture, ec=00000000
| CGMREP: Outputting: 00A0
| CGMREP: (0,5) End of Picture Frame
| CGMREP: ADMDGMD: End of Metafile, rc=00002C24
| CGMREP: Outputting: 0040
| CGMREP: (0,2) End of Metafile 0040
| CGMREP: Outputting: 00
| CGMREP: ADMDGMSV: ADMDGMD RC= 00000000
| CGMREP: ADMDGMSV: CGM Closed 00000000

| 00005617 END OF GDDM DIAGNOSTIC TRACE 11 APR 1996 2:53 PM
```

Other tracing methods

Product-sensitive programming interface

This chapter has described how to use GDDM tracing by means of TRCESTR statements in external defaults files. You can also use TRCESTR statements for GDDM tracing in the following ways:

- Specify ADMMDFT TRCESTR statements in the source of an external defaults module. (How to create such a module is described in the *GDDM System Customization and Administration* book). This module is called:
 - ADMADFC under CICS
 - ADMADFI under IMS
 - ADMADFT under TSO (and for the TSO Print Utility)
 - ADMADFV under VM
 - ADMADFD under VSE.

- In your application program, include calls to ESSUDS, specifying a source-format UDS for the ADMMDFT TRCESTR statements.

The ESSUDS statement is described in the *GDDM Base Application Programming Reference* book.

- In your application program, include calls to ESEUDS, specifying an encoded UDS for the ADMMDFT TRCESTR statements.

The ESEUDS statement is described in the *GDDM Base Application Programming Reference* book.

- In your application program, include calls to SPINIT, specifying in the systems programmer interface block (SPIB) an encoded UDS for the ADMMDFT TRCESTR statements.

The SPINIT statement and the format of the SPIB are described in the *GDDM Base Application Programming Reference* book.

Note: A single trace statement in an application call must conform to the syntax for a complete program. For example, all parts of an IF THEN ELSE statement must be within a single application call. However, a trace specification can be built up from any number of separate application calls; these are appended to the trace program formed from an external file, if there is one.

There are several ways to turn on GDDM trace facilities that involve changing the value of the *trace control word* that is held internally by GDDM. The trace control word is a fullword integer that defines the kind of trace to be produced. If you do not specify it, its value is 0 (resulting in no trace activity). Other values for the trace control word are described under “The FSTRCE statement” on page 70.

To change the trace control word for any GDDM utility or for an application program, do one of the following:

- Specify the ADMMDFT TRACE value in a source-format defaults file. (The ADMMDFT macro is described in the *GDDM System Customization and Administration* book.

Source-format defaults files are normally available only under TSO or VM systems. However, modified versions of this mechanism can also be used under CICS or VSE.

- Specify the ADMMDFT TRACE value in an external defaults module.

To change the trace control word from an application program:

- Include calls to FSTRCE, specifying the trace value as a parameter. This is described under “The FSTRCE statement.”
- Include calls to ESSUDS, ESEUDS, or SPINIT, specifying a source-format UDS for the ADMMDFT TRACE default.

The FSTRCE statement

CALL FSTRCE(*control*)

The FSTRCE statement controls internal trace functions. It is intended for internal error diagnosis.

By default, trace is deactivated.

In coding the *control* parameter for FSTRCE, you must decide the level of trace you need to diagnose the problem:

Level 1 – Component

Level 2 – Subcomponent

Level 3 – Module.

Each lower trace level includes tracing of higher levels. For example, level 3 tracing (module tracing) also includes levels 2 and 1 (subcomponent and component tracing).

Note: Depending on the complexity of the program being diagnosed, you may get excessive quantities of trace output generated if level 3 trace is used with all components enabled. Therefore, you are advised to initially use level 1 to locate the component in error, and then use level 3 with all other components disabled to enable you to trace the specific module in error.

Reminder: The first byte or bit of a sequence is designated as byte or bit 0, as appropriate.

Parameters

control (fullword integer)

A fullword integer, the trace control word, whose contents determine the type of tracing and the components to be traced, and which has the following format:

Bytes 0 and 1: flags

Bit 0	Set to 1 to make the trace word unchangeable. If this bit is set, any following calls to FSTRCE are ignored. This bit can be used to force a setting of trace (typically by using the defaults file), regardless of any FSTRCE calls in an application. This may be of use if an application already contains FSTRCE calls, and if the source of the application is not readily available.
-------	--

Bits 1—15 Component disablement flags.

Set to 1 to *disable* tracing for component **n** according to the following table:

n	Component
1	Application programming interface
2	Terminal services interface
3	Environmental services interface
4	Full screen manager
5	Presentation Graphics routines (GDDM-PGF)
6	Interactive Chart Utility (GDDM-PGF)
7	Image Symbol Editor (GDDM Base)
8	Common services interface
9	Vector Symbol Editor (GDDM-PGF)
10	Output print utility
11	IMS Interactive Utility scheduler
12	Numerical processing routines (GDDM-PGF)
13	GDDM-IMD
14	Reserved
15	Image manager.

Byte 2: trace qualifier

Can be set to produce abend dumps or additional trace output.

Unless otherwise stated, the trace qualifiers operate regardless of the component disablement flag settings and trace levels. The trace qualifiers in hexadecimal are:

- 00 No dumps: produces trace as controlled by other flags.
- 02 Abend (only if the terminal services interface component trace is not disabled): causes abnormal termination of GDDM (abend code 1405), within the module ADMLQU1x, immediately after the terminal characteristics have been determined.
- 03 Abend at termination: causes abnormal termination of GDDM (abend code 1051), within the module ADMACLP, immediately before starting normal termination.
- 10 Partial transmission trace: causes the first 16 bytes of terminal transmissions to be traced at the following points in execution:
 - For CICS, IMS, and TSO, immediately before outgoing terminal transmissions and after incoming terminal transmissions.
 - For VM, immediately after both incoming and outgoing terminal transmissions.
- 11 Full transmission trace: as for the partial transmission trace, but including the complete terminal transmissions.
- 12 Full transmission trace with control blocks: as for the full transmission trace, but including the applicable GDDM terminal control areas.
- 20 Storage use report: causes a report of how GDDM storage is acquired and freed throughout processing, to be generated at termination (FSTERM), or immediately before a GDDM-generated abend.

The report lists, for each block of storage used by GDDM, its length and address, the name of the module requesting the storage and the

associated event sequence number, and the name of the module releasing the storage and associated event sequence number.

The event sequence numbers tally with the sequence numbers generated in any accompanying module trace.

The report does not include the following items of storage:

- GDDM initial storage allocation
- GDDM dynamic save area stacks
- Storage acquired or freed during trace processing
- Storage acquired or freed while trace is not active.

Byte 3: trace level

Possible settings, in hexadecimal, are:

- 00 Stop component, subcomponent, and module trace
- 01 Start trace of component entry and exit
- 02 Start component trace plus subcomponent entry and exit
- 03 Start component and subcomponent trace plus module entry and exit.

Abend dump output

If 02 or 03 is specified as the trace qualifier, dumps are produced by abnormal termination of GDDM.

On abnormal termination of GDDM, the subsystem normally produces diagnostic messages incorporating the abend code. An exception is TSO. After receiving the message PROGRAM TERMINATED DUE TO ERROR+, you should enter a question mark (?). TSO produces a second-level message incorporating the abend code, if applicable. If a SYSABEND or SYSUDUMP file has been previously allocated, and you then press ENTER without any intervening input, TSO produces a dump.

_____ End of Product-sensitive programming interface _____

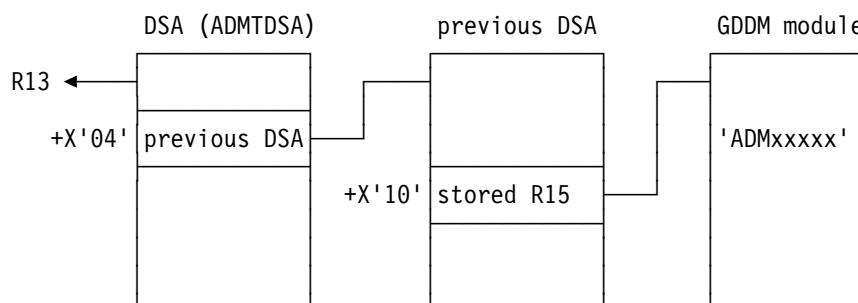
Locating control blocks and the in-storage trace table

_____ Diagnosis, Modification or Tuning Information _____

The following sections define a procedure for locating the primary GDDM control blocks, including the in-storage trace table, from a dump. This procedure can be used to locate:

- The currently active GDDM modules in the absence of trace output
- The current GDDM control blocks, including the in-storage trace table.

Locating the currently active GDDM module

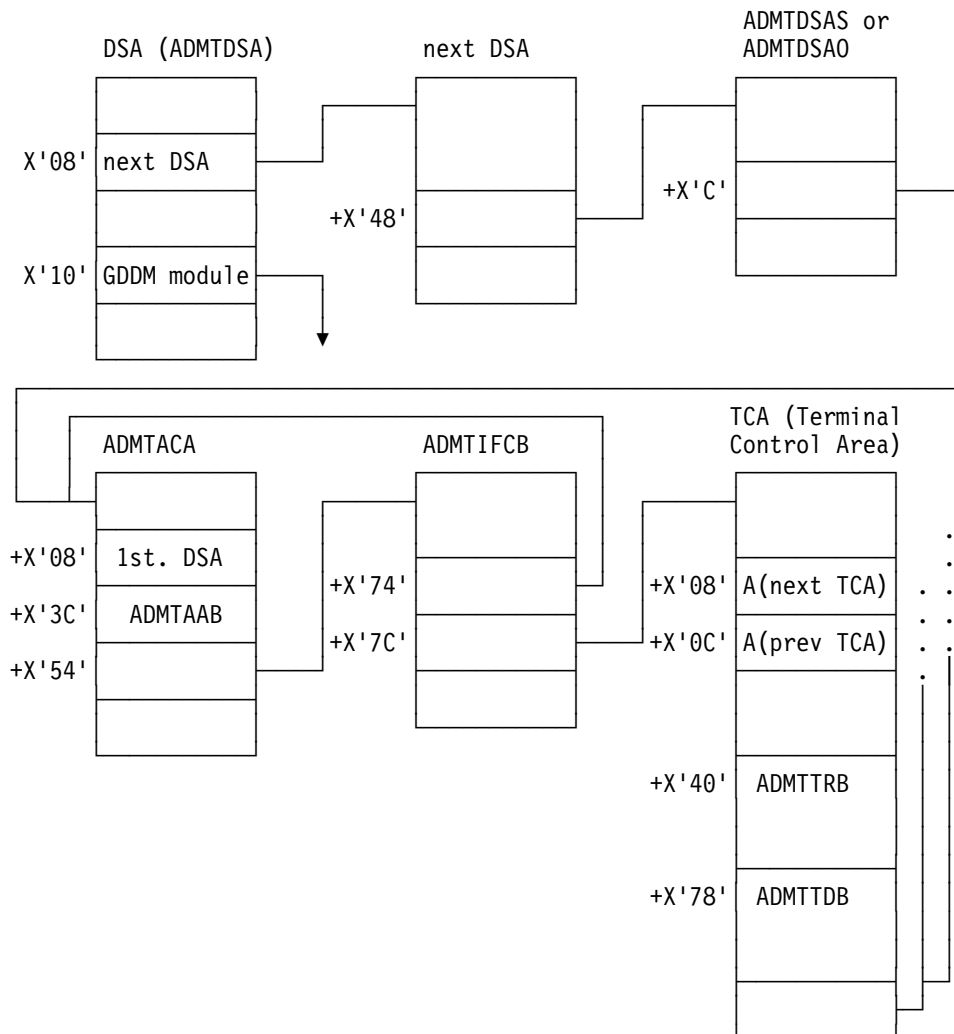


1. Check whether the abend code is listed in Appendix B, “Abend codes” on page 173. If it is listed, use the information provided in that appendix to help you locate the currently active GDDM module.
If the abend code is not listed, continue with the next step in this procedure.
2. Locate the current value of register 13. Usually, abend dumps contain a list of registers on entry to abend.
3. Locate the save area (ADMTDSA) using the address given in register 13.
Under VSE, register 4 is loaded with the address of the save area before an abend dump.
4. Locate the preceding save area using the address given at offset X'4' in the current save area.
5. Locate the executable code that has stored its registers on entry in this save area, using the address given at offset X'10' (16 decimal) in this save area (stored register 15).
6. Determine whether this executable code is a GDDM module by looking for the EBCDIC representation of the module name shortly following the start of the executable code. If it is a GDDM module, it starts with the letters ADM or AEM.
7. If the EBCDIC representation of the module name is not present or is not that of a GDDM module, return to Step 3. If Step 3 is no longer valid, go to “Locating the in-storage trace table if the save area chain is corrupted” on page 75 for another way to locate GDDM control blocks.
8. The located module should be the current GDDM module. Repeat Steps 3 through 6 to get the names of the other GDDM modules that were currently active at the time of the dump.

Locating the GDDM control blocks from the current module

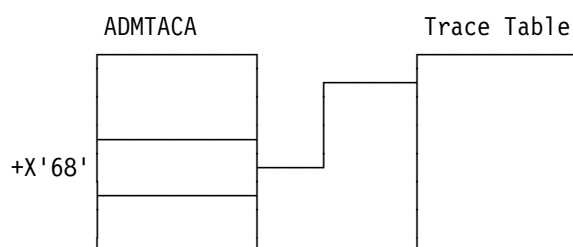
1. From a save area used to locate a currently active GDDM module, go to the *next* save area using the address given at offset X'8' in the current save area.
2. Locate the dynamic save area stack header (ADMTDSAS or ADMTDSAO), using the address given at offset X'48' (72 decimal) in this next save area.
3. Locate the application interface component control area (ADMTACA) using the address given at offset X'0C' (12 decimal) in the stack header.

4. Locate the GDDM interface control block (ADMTIFCB) using the address given at offset X'54' (84 decimal) in the AIC control area stack header. Under VSE, register 2 is loaded with the address of ADMTIFCB before an abend dump.
5. Locate a chain of terminal control areas, each including a terminal request block (ADMTTRB) and a terminal descriptor block (ADMTTDB). The address at offset X'5C' (92 decimal) in the interface control block (ADMTIFCB) gives the start of the chain of terminal control areas; each contains a terminal request block at offset X'40' (64 decimal) and a terminal descriptor block at offset X'78' (120 decimal).



Locating the in-storage trace table

Locate the in-storage trace table using the address given in field ACATRCTB at offset X'68' (104 decimal) in the ADMTACA. Note that this field is zero and no trace table is present if trace has not been enabled, or if there is not enough storage to allocate an in-storage trace table.



Locating the in-storage trace table if the save area chain is corrupted

1. If it is not possible to locate the save area chains using the mechanisms described above, scan the dump for a data area containing the characters IFCT.

This data area is the GDDM interface control table, and is part of the interface control block (ADMTIFCB). The characters IFCT are actually at offset X'4' in the IFCB.

2. Locate the application interface component control area (ADMTACA), using the address given at offset X'74' (116 decimal) in the IFCB.
3. Locate the initial GDDM save area, using the address given at offset X'8' in the ADMTACA.

Verify the interface control block (IFCB) and locate the application anchor block (AAB), using the addresses given at offsets X'54' (84 decimal) and X'3C' (60 decimal), respectively, in the ADMTACA.

4. From the initial GDDM save area, locate lower-level GDDM save areas using the forward chains at offset X'8' in these save areas. Note that, using this procedure, you can chain forward into save areas that are no longer active.
5. For each of these save areas, locate the executable code that has stored its registers on entry in the save area, using the address given at offset X'10' (16 decimal) in the save area (stored register 15).
6. Determine whether this executable code is a GDDM module by looking for the EBCDIC representation of the module name shortly following the start of the executable code. If it is a GDDM module, it starts with the letters ADM or AEM.
7. Locate the in-storage trace table, using the address given in field ACATRCTB at offset X'68' (104 decimal) in the ADMTACA. Note that this field is zero and no trace table is present if trace has not been enabled, or if there is not enough storage to allocate an in-storage trace table.

Format of in-storage trace table

In addition to being written to a trace data set, trace records are also written in an abbreviated format, into a cyclic in-storage trace table, the format of which is shown in the next diagram.

By default, the cyclic in-storage trace table contains room for the last 100 trace records. This default is defined by the ADMMDFT TRTABLE value in the external defaults module, and can be increased if required, up to a maximum of 1000. The in-storage trace table can be located as described on page 74.

ADMTACA		
ACATRCTB	ADMTTRTB	In-storage-trace-table
Hex offsets	0	'GDDM TRACE TABLE'
	10	Length of Trace Table
	14	A(first entry in table)
	18	A(first entry in table)
	1C	A(next entry to be built
	20	first trace entry in table
	40	next entry in trace table

100 (default)	xx	latest trace entry in table
trace entries	xx	'CURRENT ENTRY. OLDEST FOLLOWS. '
each of length	xx	-----
32 bytes	xx	oldest trace entry in table

		last trace entry in table
		'END OF GDDM TRACE TABLE'

Meaning of in-storage trace records

Each in-storage trace record is 32 bytes in size, with two possible formats, depending on the trace record type. The trace record type is identified by bytes 6 and 7 of the trace record.

Format for CPNIN, CPNOUT

Bytes	Description
00—03	Event Sequence Number (binary)
04—05	DSA Level (EBCDIC)
06—07	Trace Record Type (EBCDIC): '10' CPNIN (component entry) '11' CPNOUT (component exit)
08	Blank
09—14	Request control parameter (RCP) mnemonic (EBCDIC)
15	Blank
16—19	Request control parameter (RCP) code (binary)

20—28 Error message identifier and severity (EBCDIC)
 29—31 Error message type (EBCDIC):
 '(I)' Internal error code
 '(E)' External error code

The meaning of each field is as described in “Format of trace output” on page 42.

The following is an example of a CPNOUT trace record, in EBCDIC:

```
'xxxx0111 FSPCRT yyyyADM0130 E(E)'
```

This is analyzed as follows:

Bytes	Contents	Meaning
00—03	xxxx	Event sequence number xxxx (in binary)
04—05	'01'	DSA level 1
06—07	'11'	CPNOUT (component exit)
09—14	FSPCRT	GDDM function is FSPCRT
16—19	yyyy	RCP code (in binary)
20—26	ADM0130	Error feedback contains error code 130
28	E	Error feedback is severity 'E'
29—31	(E)	Error feedback represents an external error.

Format for SUBIN, SUBOUT, MODIN, MODOUT

Bytes	Description
00—03	Event sequence number (binary)
04—05	DSA level (EBCDIC)
06—07	Trace record type (EBCDIC): '20' SUBIN (subcomponent entry) '21' SUBOUT (subcomponent exit) '30' MODIN (module entry) '31' MODOUT (module exit)
08—31	Module identification, including module name.

The meaning of each field is as described on page 42.

The following is an example of a MODOUT trace record, in EBCDIC:

```
'xxxx0131ADMACFP 96100 V3R2.0 '
```

This is analyzed as follows:

Bytes	Contents	Meaning
00—03	xxxx	Event sequence number xxxx (in binary)
04—05	'01'	DSA level 1
06—07	'31'	MODOUT (module exit)
08—31	aaaa	Module identification, including module name, ADMDSRO

_____ End of Diagnosis, Modification or Tuning Information _____

Chapter 3. IKT0405I debugging procedure

This chapter applies only to users running TSO under MVS.

If a data stream error occurs when GDDM tries to write to a display screen, the TSO message IKT0405I may be displayed. This chapter describes the procedure for debugging errors of this kind. It describes how to enable the GDDM trace, how to start the Generalized Trace Facility (GTF), and how to start the VTAM trace before you recreate the error. It then describes how to initiate a dynamic dump. You are then told how to stop tracing and how to format the trace. Finally, there is a section on how to locate the failing component.

Starting the traces

You must enable the GDDM trace and then start GTF and the VTAM trace.

Enabling the GDDM trace

To enable the GDDM trace, use the FULLIO parameter in the ADMMDFT TRCESTR statement in the ADMDEFS PROFILE. The FULLIO parameter will, in most cases, give sufficient trace information. For details of the other functions available with the TRCESTR keyword and the trace data that they provide, refer to Chapter 2, "GDDM tracing" on page 21.

Once enabled, the GDDM trace starts automatically when the process to be traced is started.

Starting the Generalized Trace Facility for data stream errors

You can start the Generalized Trace Facility (GTF) from the system console or from the SDSF log. From the system console, enter the following commands:

```
S GTF.STE
nn TRACE=SVCP,USR,JOBNAMEP
nn SVC=(93,94)
nn JOBNAME=tsouserid
nn END
nn U
```

Where:

1. The STE parameter in the START (S) command is an identifier used later to stop GTF.
2. The nn parameters are reply identifiers used when replying to a sequence of GTF messages.
 - "nn TRACE= SVCP,USR,JOBNAMEP" requests tracing of selected SVCs and limits the tracing to the specified job or TSO userid.
 - "nn SVC=(93,94)" specifies the SVCs to be traced.
 - "nn JOBNAME=tsouserid" specifies the TSO userid to be traced.
 - "nn END" indicates the end of the user specifications.
 - "nn U" starts GTF tracing.

If the commands to start GTF are issued from the SDSF environment, you must precede each line by a slash (/).

Starting the VTAM trace

You can start the VTAM trace from the system console or from the SDSF log by issuing the following commands:

```
F VTAM,TRACE,TYPE=TSO,ID=tsouserid
F VTAM,TRACE,TYPE=BUF,ID=terminal_luid
```

Notes:

1. You must specify the parameter CONFTXT=N0 in the TSOKEYnn member in SYS1.PARMLIB, or the trace will suppress the TSO buffers with the message CONFIDENTIAL AND SUPPRESSED. Any change to this parameter takes effect at the next logon.
2. In the above examples "VTAM" stands for the name of the address space in which VTAM is running. An alternative name is NET.
3. tsouserid is the TSO userid.
4. terminal_luid is the terminal id.

Recreate the error

After starting the traces, you should recreate the error. You should also take a dynamic dump of the TSO address space in error at the point where the message IKT0405I followed by three asterisks (***) appears on the display screen.

To make it easier to debug the VTAM trace, you are recommended to recreate the error from a different TSO userid, or from the console.

The dynamic dump

You should take a dynamic dump if you think that the task is in a loop, or in the WAIT state, or if the IBM Support Center staff ask you for a dump to enable them to relate addresses in a GTF trace to modules in the TSO user's address space.

You can initiate a dynamic dump with the DUMP operator command. The dump is written to any free SYS1.DUMPnn data set, where nn is a numeric variable.

You can check that there is a free SYS1.DUMPnn data set by entering a DISPLAY DUMP STATUS command from the console.

The format of the command is:

```
D D,ST
```

or

```
D D,T
```

where the parameters are defined as follows:

STATUS (ST or S)

The system is to display a summary of the full or available status of each defined SYS1.DUMPnn data set. This information is displayed in two sections, one for DASD data sets and the other for tape data sets.

TITLE (T)

The system is to display the dump title and time of dump for each full direct access dump data set that you specify on the DSN parameter. This information is not displayed for tape dump data sets.

If all the SYS1.DUMPnn data sets are full, you must use the DUMPDS command to clear one or more of them. The format of the command is:

```
DD CLEAR,DSN=nn
```

where nn is the number of the SYS1.DUMP data set.

You can now initiate the dynamic dump by entering the DUMP command. The format of the command is:

```
DUMP COMM=('This is the title for the dump')
```

The title you give the dump can be 1 to 100 characters long. This title becomes the first record in the dump data set.

In response to the DUMP command, the system prompts you for the dump options with the following message:

```
nn IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

where nn is the message reply number.

You must respond to the IEE094D message with REPLY commands specifying dump options.

1. First specify the address space to dump using the ASID number, the jobname, or the TSO userid. The formats of the corresponding REPLY commands are:

```
nn,ASID=3E,CONT
```

or

```
nn,JOBNAME=JES2,CONT
```

or

```
nn,TSOname=USERID,CONT
```

Note: In the above examples, the parameter nn is the reply number of the IEE094D message.

The CONT parameter at the end of the command indicates to the dump command that you are going to specify additional parameters for the dump. This results in a further IEE094D message.

2. You must reply to this IEE094D message with a parameter SDATA specifying which areas of the system should be contained in the dump. The minimum SDATA specification is as follows:

`mm,SDATA=(CSA,GRSQ,RGN,PSA,SQA,TRT),END`

where `mm` is the reply number of the IEE094D message to which you are responding.

Note: The END parameter indicates to the dump command that no further dump parameters are to be specified.

The SDATA options shown represent the following system areas:

CSA	Common Service Area
GRSQ	Global Resource Serialization (ENQ/DEQ/RESERVE) Queues
PSA	Prefixed Storage Area for all processors
RGN	Private Area of address space being dumped,including LSQA and SWA
SQA	System Queue Area
SUM	Summary Dump
TRT	GTF, system trace, master trace, and NIP hardcopy buffer data.

Note: Always check the console for an IEA911E message. If this message is displayed, only a partial dump has been written to the allocated data set, which might be unusable for later debugging. If IEA911E is displayed, you must increase the size of the dump data set, and recreate the error.

Stopping the traces

When you have finished tracing, you must disable GDDM tracing and stop the GTF and VTAM tracing.

Disable GDDM tracing

You can disable GDDM tracing by erasing the TRCESTR statements from your default file, or by turning them into comments. To comment them out, place an asterisk (*) in column 1 of each TRCESTR default statement.

Stop GTF

To stop GTF you must issue the command:

`P STE`

where STE is the identifier used in the START GTF command in step 1 on page 79.

Stop VTAM tracing

To stop VTAM tracing, enter the following commands from the SDSF log or the master console:

```
F VTAM,NOTRACE,TYPE=TSO,ID=tsouserid
```

```
F VTAM,NOTRACE,TYPE=BUF,ID=terminal_luid
```

Note: In the above examples “VTAM” stands for the name of the address space in which VTAM is running. An alternative name is NET.

Format the trace

To format the GTF output, use an IPCS GTFTRACE statement.

How to locate the failing component

To identify the failing component, follow these steps:

1. Look in the VTAM trace for the IKT0405I message.
2. Find the nonzero reply from the terminal in the VTAM INBOUND entries preceding the IKT0405I message.
3. From this entry, identify the OUTBOUND buffer for which a nonzero return was reported.
4. Just before these VTAM buffers (one for the line and one for the user) you should see an SVC 93 command which was used to send the contents of the buffer to the terminal.
5. Note the PSW NSI address of this SVC 93 and use the address in the dynamic dump to identify the module from which the SVC was issued.
6. This identifies the component in control of the failing buffer sent to the terminal, and provides you with a good starting point for problem analysis.

Chapter 4. GDDM Interactive Map Definition diagnosis

This chapter describes an additional diagnostic facility that is supplied to help solve problems in GDDM Interactive Map Definition (GDDM-IMD). It is primarily intended for use under the direction of IBM Support Center staff.

Using the GDDM-IMD diagnostic facility

You can use the GDDM-IMD diagnostic facility for two purposes:

- To display the contents of main storage
- To trace and check operations performed on the map specification library (MSL).

You can display the contents of main storage at any time during GDDM-IMD operation.

You can turn the MSL trace on and off at any time during GDDM-IMD operation. GDDM-IMD automatically displays the applicable information whenever the MSL is accessed. If GDDM trace is active, the information is also sent to the GDDM trace file.

When you report a failure in GDDM-IMD, the IBM Support Center staff may ask you to repeat the operation that failed, with the MSL traces or record validation turned on. You should direct the output this generates to the GDDM trace file by turning on the GDDM trace facility.

If you are asked to submit an APAR, include any messages generated by record validation in the ERR-DESCRIPTION field in the APAR data set.

Invoking the GDDM-IMD diagnostic facility

To invoke the GDDM-IMD diagnostic facility:

1. Press the HELP key (PF1) to start the tutorial.
2. Enter the DEBUG command (or D) in the command area of the tutorial frame. The GDDM-IMD diagnostic frame is then displayed.

Note: You can use the DEBUG command even if the tutorial is not available.

The GDDM-IMD diagnostic frame

The example below shows the GDDM-IMD diagnostic frame. When this frame is displayed, you can select the functions of the GDDM-IMD diagnostic facility that you require. You can:

- View the contents of main storage associated with the operation you currently have running in GDDM-IMD. (This facility can be restricted to specified users at your enterprise.)
- Turn on and off the MSL access trace, the MSL record trace, and record validation.

DB -----GDDM-IMD DIAGNOSTICS ----- DEBUG

```

MSL ACCESS TRACE      ==> NO
MSL RECORD TRACE      ==> NO      RECORD VALIDATION ==> NO
STORAGE TO DISPLAY    ==> ECT

```

```

* 2A558 C4C5C3E3 00009460 0002A798 00000000 *DECT,,m-,,xq,,,*,
2A568 0002A990 0002A794 000291B8 0002BE30 *,,z,,,xM,,j,,,*,
2A578 F2040054 00000000 FFFFFFFF 0002A731 *2,,,,,,,,,,,,,X,*,
2A588 0200F404 0002A96C 00000000 00000000 *,,4,,,z%,,,,,,,,*,

2A598 00000000 00000000 F1000000 01004000 *,,,,,,,,,,1,,,,,*,
2A5A8 00000000 E2C1D4D7 D3C50000 D4C1D700 *,,,,SAMPLE,,MAP,*,
2A5B8 00000000 C4F50000 00000006 0003710C *,,,,D5,,,,,,,,,*,
2A5C8 0002BF18 00036348 40400020 0050FFFF *,,,,,,,,,, ,,&,,*,

2A5D8 FFFEFFFF FFFF0080 00029188 0002B5FC *,,,,,,,,,,jh,,,,*,

2A5E8 00000000 00000000 4E615F7B 00180050 *,,,,,,,,,,+/-#,,,a*,
2A5F8 04007C6C 00000000 0002C094 00000000 *,,0%,,,,,(m,,,,*,
2A608 00000000 0002B834 00000000 00010050 *,,,,,,,,,,,,,,a*

```

USE END KEY (PF3) TO TERMINATE

Main storage display

The main storage display facility is available only if one of the 31 low-order bits of the GDDM trace control word is set to 1. For details of setting the GDDM trace control word, see Chapter 2, "GDDM tracing" on page 21.

The first column shown in each line of the storage display contains a hexadecimal address. The other columns of the display show the contents of the 16 bytes of storage starting at that address, in hexadecimal and in characters. Characters that are not displayed are shown as periods (.). The storage address is unprotected and you can update it by typing over it so that a specified area of storage is displayed.

You can set the starting address of the storage to be displayed in the STORAGE TO DISPLAY field at the top of the frame. You specify the address as a hexadecimal expression, which can contain the following operators and special values:

- + plus
- minus
- % designates that the expression to the left of the percent symbol is an address containing the fullword value to be used. (See the examples below.)
- * designates the address of the storage in the current display.

Example addresses

The following address:

1814%+8%

means: take the value of the fullword at address X'1814' and add 8 to it. The result is an address that in turn contains a fullword. This value of this fullword is the address of the storage to be displayed.

The following address:

*-C8

means: display the storage starting 200 bytes before the start of the currently displayed storage.

You can use PF8 and PF7 (or PF20 and PF19) to scroll the current storage display forward and backward by its current length. If you place the cursor under any of the hexadecimal fields in the storage display and press PF9 (or PF21), the contents of the field are used as the starting address of a new area of storage to be displayed.

To resume normal GDDM-IMD operation, press PF3.

Note: If you try to display storage to which GDDM-IMD does not have read access, an abnormal termination occurs in module AEMDIA81 for the unacceptable address.

Tracing MSL operations

The GDDM-IMD diagnostic frame contains three input fields to control how the map specification library (MSL) is traced:

- MSL ACCESS TRACE controls the display of MSL access requests (open, get, put, close, and so on)
- MSL RECORD TRACE controls the display of MSL records
- RECORD VALIDATION controls the validation of records.

In each case, enter YES (or Y) to turn the facility on, or NO (or N) to turn it off. The initial value of each is NO.

Press PF3 to resume normal GDDM-IMD operation.

Using the MSL traces

If MSL traces are turned on, the GDDM-IMD screen is cleared every time there is trace data to show, and the trace data is displayed. When the screen is full, or there is no more trace data to be displayed, the following prompt appears:

AEM00201A PRESS ENTER TO CONTINUE, OR CANCEL

Pressing any interrupt key causes the next frame of trace data, or the next GDDM-IMD frame, to appear. Pressing the cancel key (PF6/18) stops the display of any trace data generated before the next GDDM-IMD frame. However, if the GDDM trace facility is active, the trace data is still directed to the GDDM trace data set.

The MSL access trace

The MSL access trace displays invocations of the MSL access instructions with the options that they specify. Here is a typical access trace:

```

HEDDBOPN  SAMPLE_____
MSL=00/01 OBJ=02/02 LVL=03/03 ACC=UPD STA=A NXT=N CPY=D DIR=000F7FFC RC=0001
HEDDBPUT  SAMPLE_____
MSL=01 REC=F304 AT=0004DD08 1ST=0001 LST=0001 BRK=N RTN=Y MOF=Y
DIR=000F7FFC RC=0000
HEDDBGET  SAMPLE_____
MSL=01 REC=F304 AT=000F980C 1ST=0001 LST=0001
DIR=000F7FFC RC=0000
HEDDBOPN  SAMPLE__ MAP_____
MSL=00/01 OBJ=01/01 LVL=03/03 ACC=INP STA=0 NXT=G CPY=D DIR=000FB4FC RC=0000
HEDDBGET  SAMPLE__ MAP_____
MSL=01 REC=F304 AT=000FC50C 1ST=0001 LST=FFFF
DIR=000FB4FC RC=0000
HEDDBFRE  SAMPLE__ MAP_____
MSL=01 REC=F304 DIR=000FB4FC RC=0000
HEDDBCLS  SAMPLE__ MAP_____
MSL=01 RC=0000
HEDDBOPN  SAMPLE__ MAP_____
MSL=00/00 OBJ=01/00 LVL=03/00 ACC=INP STA=0 NXT=G CPY=D DIR=00000000 RC=0202

AEM00201A PRESS ENTER TO CONTINUE, OR CANCEL

```

The MSL record trace

The MSL record trace displays the MSL data records as they are read from the MSL and written to the MSL. Here is an example of the output produced by the MSL record trace:

```

HEDDBGET  SAMPLE__ MAP_____
MSL=01 REC=F304 AT=000C01CC 1ST=0001 LST=FFFF
00260000 000C01F4
0008      01000000 00180050 00010001 00000000 *.....&.....*
0018      00004040 40404040 40408300 0000      *..          C... *
00140000 000C0208
0008      02007B5F 6C614E7C 00010001      *..#~%/+@.... *
00330001 00000000
0008      03000003 C1000000 000000E0 00000000 *....A.....*
0018      00D70000 00000000 60000000 0000E400 *.P.....-.....U.*
0028      00000000 00200000 000000      *.....      *
DIR=0009427C RC=0000
HEDDBPUT  SAMPLE__ MAP_____
MSL=01 REC=F304 AT=000C01CC 1ST=0001 LST=0004 BRK=N RTN=Y MOF=Y
00260000 000C01F4
0008      01000000 00180050 00010001 00000000 *.....&.....*
0018      00004040 40404040 40408300 0000      *..          C... *
00140000 000C0208
0008      02007B5F 6C614E7C 00010001      *..#~%/+@.... *
00330801 000C6790 S
0008      03000003 C1000000 000000E0 00000000 *....A.....*
0018      00D70000 00000000 60000000 0000E400 *.P.....-.....U.*
0028      00000000 00200000 000000      *.....      *
00400501 00000000 RF
0008      03000004 C1000000 000000E0 00000000 *....A.....*
0018      00C20000 00000000 70000000 0000D700 *.B.....-.....P.*
0028      00000000 00600000 000000E4 00000000 *.....-.....U....*
0038      00002000 00000000      *.....      *
DIR=0009427C RC=0000

AEM00201A PRESS ENTER TO CONTINUE, OR CANCEL

```

Record validation

If record validation is turned on, the diagnostic facility checks the records sent to and received from the MSL. If any check fails, trace messages are generated. If the MSL record trace is also turned on, such messages are also displayed *before* the records are traced. This list shows the messages that the record validation can produce:

E01 INVALID OPCODE, DDBLROPC=xx

Reason: The MSL operation code is not valid.

E02 LEAVES A GAP, DPRSLINE=xxxx, DPRSCOL=xxxx

Reason: The record does not follow immediately after the previous record.

E03 NEW LINE NOT COLUMN 1, DPRSCOL=xxxx

Reason: The new line does not start in column one.

E04 NON-ZERO FLID FOR CONSTANT FIELD

Reason: A constant field appears to be linked to an application structure.

E05 ATTRIBUTE-BITS INVALID

Reason: The attribute field (DPRSAPSK) contains an invalid bit value.

E06 JUSTIFY-BITS INVALID

Reason: Both justification bits are set.

E07 INTENSITY-BITS INVALID

Reason: The intensity attribute bit setting is not valid.

I02 LEADING/TRAILING BLANKS IN TEXT

Reason: There are leading or trailing blanks in DPRSTEXT.

S01 MISALIGNED, ADDR=xxxxxxxx

Reason: The record must be on a fullword boundary.

S02 OVERLAPS PRECEDING FIELD, DPRSLINE=xxxx, DPRSCOL=xxxx

Reason: The record overlaps the end of the previous record.

S03 LINE LENGTHS INCONSISTENT, PREVIOUS=xxxx, MAX=xxxx

Reason: The length of the previous line of the map is not the same as the maximum found.

S04 TEXT OVERFLOWS FIELD, DPRSTXTL=xx, DPRSTXTO=xx, DPRSTLEN=xx

Reason: The text offset and length are longer than the field length.

S05 INCONSISTENT LENGTHS, DDBLRLT=xxxx, DPRSTXTL=xx, DPRSXLEN=xx

Reason: The length in the logical record header is not consistent with the lengths of the component parts of the record.

S06 TOO SHORT, DDBLRLT=xxxx

Reason: The record is shorter than the maximum.

W03 CURSOR FLAG ON AGAIN

Reason: The cursor is set in more than one record.

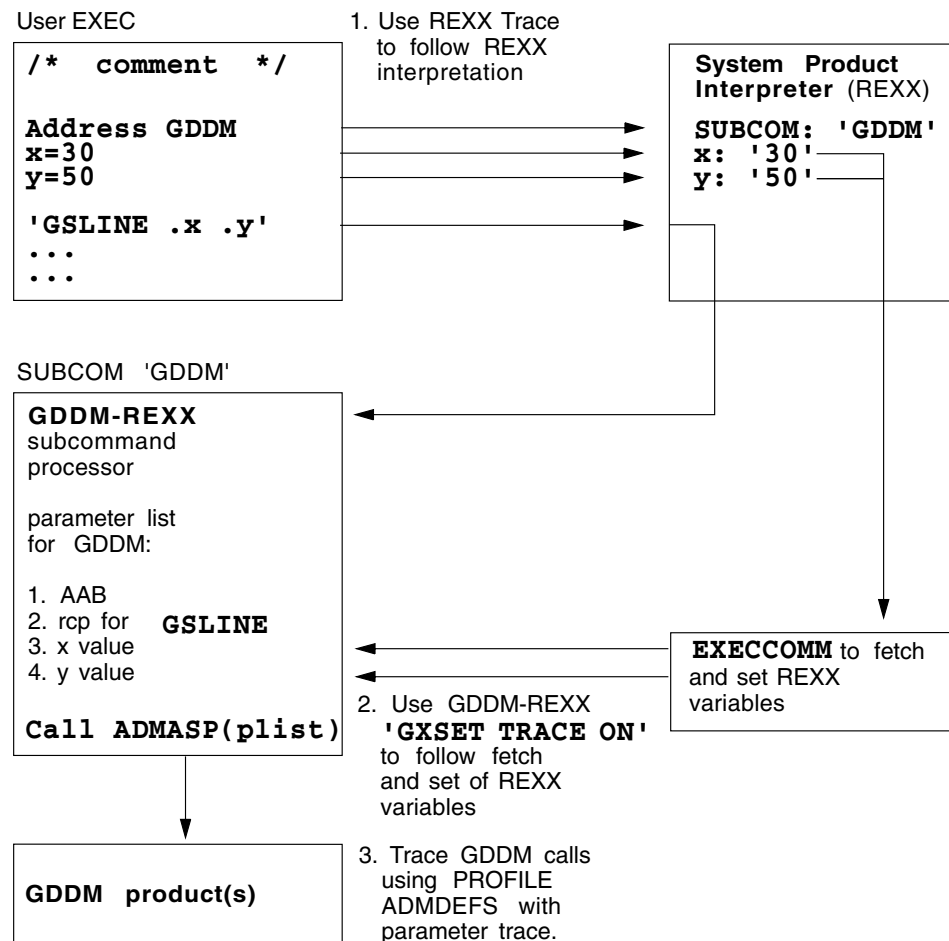
W04 RESERVED FIELD NOT NULL, DPRSTYPS=xx

Reason: A reserved bit in DPRSTYPS is not cleared to zero.

Chapter 5. GDDM-REXX diagnosis

GDDM-REXX runs in the subcommand environment of the System Product Interpreter (REXX). Any command that is not recognized by REXX is passed to the active subcommand environment. To make GDDM-REXX the active subcommand environment, the “Address GDDM” instruction is used. Any symbolic parameters passed are resolved by GDDM-REXX, which then passes calls to GDDM.

When you are trying to find the source of an error, there are three different types of tracing that you can use: REXX, GDDM-REXX, and GDDM. The following figure shows how they relate to one another.



REXX and GDDM-REXX tracing are described in this chapter. Refer to Chapter 2, “GDDM tracing” on page 21 for details of GDDM tracing.

GDDM-REXX under MVS

For information about the REXX Address instruction and REXX tracing under TSO, see *TSO/E Version 2 Procedures Language MVS/REXX Reference*, SC28-1883.

REXX and GDDM-REXX tracing under TSO

Here is part of a simple REXX EXEC which uses REXX and GDDM-REXX tracing:

```
Trace r                               /* Start REXX tracing          */
Address link 'GDDMREXX INIT'
Address gddm
s='abcde'
x=70
y=60
'GXSET TRACE ON TIME'                 /* Start GDDM-REXX tracing    */
'GSCHAR 50 50 5 .s'
'GSMOVE 40 60'
'GSLINE .x .y'
'GXSET TRACE OFF'                     /* Stop GDDM-REXX tracing    */
'ASREAD . . .'
Address link 'GDDMREXX TERM'
Trace off                             /* Stop REXX tracing         */
Exit
```

Part of the trace output appearing on the screen is shown here:

```
4 ** Address command 'GDDMREXX INIT'
  >>> "GDDMREXX INIT"
5 ** Address gddm
6 ** s='abcde'
  >>> "abcde"
7 ** x=70
  >>> "70"
8 ** y=60
  >>> "60"
9 ** 'GXSET TRACE ON TIME'             /* Start GDDM-REXX tracing
*/
  >>> "GXSET TRACE ON TIME"
ERX0000 I TIME STAMP: 08/24/88 11:15:30.119222
ERX0000 I "GXSET TRACE ON TIME"
10 ** 'GSCHAR 50 50 5 .s'
  >>> "GSCHAR 50 50 5 .s"
ERX0000 I Var fetch: s = "abcde"
ERX0000 I TIME STAMP: 08/24/88 11:15:33.211771
ERX0000 I "GSCHAR 50 50 5 .s"
11 ** 'GSMOVE 40 60'
  >>> "GSMOVE 40 60"
ERX0000 I TIME STAMP: 08/24/88 11:15:33.271971
ERX0000 I "GSMOVE 40 60"
12 ** 'GSLINE .x .y'
  >>> "GSLINE .x .y"
ERX0000 I Var fetch: x = "70"
ERX0000 I Var fetch: y = "60"
ERX0000 I TIME STAMP: 08/24/88 11:15:33.284259
ERX0000 I "GSLINE .x .y"

1 13 ** 'GXSET TRACE OFF'             /* Stop GDDM-REXX tracing
*/
  >>> "GXSET TRACE OFF"
14 ** 'ASREAD . . .'
  >>> "ASREAD . . ."
15 ** Address command 'GDDMREXX TERM'
  >>> "GDDMREXX TERM"
```

```
16 ** Trace off
```

```
/* Stop REXX tracing
```

GDDM-REXX under VM

For information about the REXX Address instruction and REXX tracing under VM, see:

VM/ESA Procedures Language VM/REXX Reference, SC24-5466

REXX and GDDM-REXX tracing under VM

Here is part of a simple REXX EXEC which uses REXX and GDDM-REXX tracing:

```
Trace r                               /* Start REXX tracing          */
Address command 'GDDMREXX INIT'
Address gddm
s='abcde'
x=70
y=60
'GXSET TRACE ON TIME'                 /* Start GDDM-REXX tracing    */
'GSCHAR 50 50 5 .s'
'GSMOVE 40 60'
'GSLINE .x .y'
'GXSET TRACE OFF'                     /* Stop GDDM-REXX tracing    */
'ASREAD . . .'
Address command 'GDDMREXX TERM'
Trace off                             /* Stop REXX tracing          */
Exit
```

The output from the REXX and GDDM-REXX traces was spooled using the CP command SPOOL CONSOLE START. Here is part of that output:

```
4 ** Address command 'GDDMREXX INIT'
   >>> "GDDMREXX INIT"
5 ** Address gddm
6 ** s='abcde'
   >>> "abcde"
7 ** x=70
   >>> "70"
8 ** y=60
   >>> "60"
9 ** 'GXSET TRACE ON TIME'             /* Start GDDM-REXX tracing
*/
   >>> "GXSET TRACE ON TIME"
ERX0000 I TIME STAMP: 08/24/88 11:15:30.119222
ERX0000 I "GXSET TRACE ON TIME"
10 ** 'GSCHAR 50 50 5 .s'
   >>> "GSCHAR 50 50 5 .s"
ERX0000 I Var fetch: s = "abcde"
ERX0000 I TIME STAMP: 08/24/88 11:15:33.211771
ERX0000 I "GSCHAR 50 50 5 .s"
11 ** 'GSMOVE 40 60'
   >>> "GSMOVE 40 60"
ERX0000 I TIME STAMP: 08/24/88 11:15:33.271971
ERX0000 I "GSMOVE 40 60"
12 ** 'GSLINE .x .y'
   >>> "GSLINE .x .y"
ERX0000 I Var fetch: x = "70"
ERX0000 I Var fetch: y = "60"
ERX0000 I TIME STAMP: 08/24/88 11:15:33.284259
ERX0000 I "GSLINE .x .y"

1 13 ** 'GXSET TRACE OFF'             /* Stop GDDM-REXX tracing
```

GDDM-REXX diagnosis

```
*/  
    >>> "GXSET TRACE OFF"  
14 ** 'ASREAD . . .'  
    >>> "ASREAD . . ."  
15 ** Address command 'GDDMREXX TERM'  
    >>> "GDDMREXX TERM"  
16 ** Trace off                               /* Stop REXX tracing
```

Chapter 6. GDDM-PCLK diagnosis

If IBM personal computer system users have a problem running GDDM-PCLK, they must report the problem to you.

Note: Personal computer system users *cannot* report the problem directly to the IBM Support Center staff. Try to solve any problems yourself by referring to the section “Common errors with GDDM-PCLK” before contacting the IBM Support Center staff.

If you cannot solve the problem yourself, you can call the IBM Support Center, quoting the component identifier for the GDDM Base system on which GDDM-PCLK is installed. For details on which component identifier applies to your system, and for any general points about defining problem symptoms to the IBM Support Center staff, see Chapter 8, “Reporting GDDM problems to IBM” on page 135.

This chapter tells you:

- How to recover from common errors with GDDM-PCLK
- How service is applied to GDDM-PCLK
- How to find the level of GDDM-PCLK code
- How to use GDDM-PCLK trace support
- How to ensure that any changes you have made to the host default image symbol set have been reflected in GDDM-PCLK
- How to debug Personal Communications/3270 problems

Common errors with GDDM-PCLK

There are three common errors that a personal computer system user may have with GDDM-PCLK:

1. The user sees the following message displayed

```
ADM0275 W  GRAPHICS {(IMAGE)} CANNOT BE SHOWN. REASON CODE n
```

- | | |
|--------------------|--|
| Explanation | <p>A page containing graphics, or image if this is indicated by the message, is being sent to a GDDM-PCLK-supported personal computer system with the GDDM-PCLK program available. This can be caused by one of the following situations:</p> <ul style="list-style-type: none"> • The user has not specified the GDDM-PCLK procopt to GDDM. • The user has specified the GDDM-PCLK procopt to GDDM but has also specified one of the CICS procopts BMSCOORD or PSCNVCTL. GDDM-PCLK is not compatible with these CICS procopts. • When the personal computer system was opened by the host application program, the user pressed ENTER instead of hot-keying to the personal computer system session and starting the GDDM-PCLK program. More |
|--------------------|--|

information about this problem can be found in the description of message ADM0873 in the *GDDM Messages* book.

The reason code *n* depends on the terminal emulator that you are using.

The construction of the graphics picture, or image, if this is indicated by the message, is suppressed, and the area of the display where the graphics or image should appear remains blank.

How to recover

- You must ensure that the GDDM-PCLK procopts have been set up correctly.
- The user must hot-key to the personal computer system session instead of pressing ENTER.

Note: Message ADM0275 may be displayed for several other reasons. See the *GDDM Messages* book for details.

2. The user's graphics don't look right

If the user's application deletes text that has been drawn on top of graphics, a black rectangle may be left where the text was originally drawn.

Explanation For some display adapters, GDDM-PCLK uses the same memory to hold graphics and text. If some text is deleted, the graphics from the same area of the personal computer system screen are also lost, which leaves a black rectangle on the screen.

How to recover The user must direct the application to redraw the graphics by following one of these steps:

- For many programs, the user can press the Clear key.
- For some terminal emulators, the user can press Ctrl+F5.

Sometimes the display of graphics may be incorrect with one or more of the following symptoms:

- The graphics screen is corrupted by random pixels or blocks of color.
- The graphics are white and shown only as regularly spaced vertical slices.
- The colors are not correct.

Explanation When using a terminal emulator, the user has hot-keyed to the host session without first pressing Ctrl+F9. The personal computer system screen must be clear of graphics before the user hot-keys to the host session.

How to recover The user must follow these steps:

- a. Hot-key to the personal computer system session if not already there.
- b. Press Ctrl+F9 to clear the graphics.
- c. Hot-key to the host session.
- d. Redisplay the graphics from the host session.

For many programs, pressing the Clear key causes a refresh of the graphics on the screen.

Another way to recover is for the user to stay in the personal computer system session and, using the GDDM-PCLK-supplied host key equivalents, follow these steps:

- a. Hot-key to the personal computer system session if they are not already there.
- b. Press Ctrl+F9 to clear the graphics.
- c. Press Ctrl+F9 to return to the graphics.
- d. Stay in the personal computer system session, and redisplay the graphics from the host program. Use the GDDM-PCLK-supplied host key equivalents as required by the host program. For many programs, pressing the Alt+F2 keys (Clear) causes the host program to redisplay the graphics.

3. The user's workstation disk fills up during automatic data transfer

Users may see the following message displayed when the automatic download of GDDM-PCLK from the host to their personal computer system is taking place:

GQD0410 Disk full

Explanation The automatic download process does not delete the old version of GDDM-PCLK until the new version has been downloaded. This can cause the workstation disk to become full before the process is complete.

How to recover The user must release as much space as possible, for example by deleting unwanted plot or print files. They can do this from the GDDM-PCLK panels, or by entering DOS commands in the DOS command area. After releasing the space, the user must start GDDM-PCLK again.

If the diskette still fills up, the user should follow these steps:

- a. Copy the file pclk.exe from the \PCLK11 subdirectory to another drive.
- b. Delete pclk.exe from the \PCLK11 subdirectory.
- c. With the drive that contains the \PCLK11 subdirectory as the default drive, run pclk.exe in the other drive.
- d. Choose option 1 from the GDDM-PCLK main menu. A new version of pclk.exe is automatically downloaded from the host to the \PCLK11 subdirectory.
- e. Delete the copy of pclk.exe that was made in step 3a.

Applying service to GDDM-PCLK

Service for GDDM-PCLK is always applied to the GDDM Base program. When a personal computer system user starts GDDM-PCLK host application support (option 1 from the GDDM-PCLK Main Panel), the service files are downloaded automatically from the host computer to the personal computer system on which GDDM-PCLK is installed.

Finding the level of GDDM-PCLK code

You may need to know the level of GDDM-PCLK installed on the personal computer system. Find this out by entering the command:

```
pclklev
```

in the \PCLK11 subdirectory prompt. GDDM-PCLK displays a list of the files that constitute GDDM-PCLK, with level numbers that you can quote to the IBM Support Center staff. The level numbers tell them if any service has been applied.

GDDM-PCLK trace support

For some problems, the IBM Support Center staff may ask you to investigate further. If so, you may need to run a GDDM-PCLK trace.

Note: You may also need to run a GDDM trace in conjunction with the GDDM-PCLK trace. See Chapter 2, “GDDM tracing” on page 21 for details on how to do this.

To run a GDDM-PCLK trace, follow these steps:

1. Access GDDM-PCLK trace from option 5 “Service functions” on the GDDM-PCLK Main Panel.
2. To make option 5 appear, run GDDM-PCLK by entering the command `pclk /s`, where `/s` means service.

The GDDM-PCLK Main Panel is displayed:

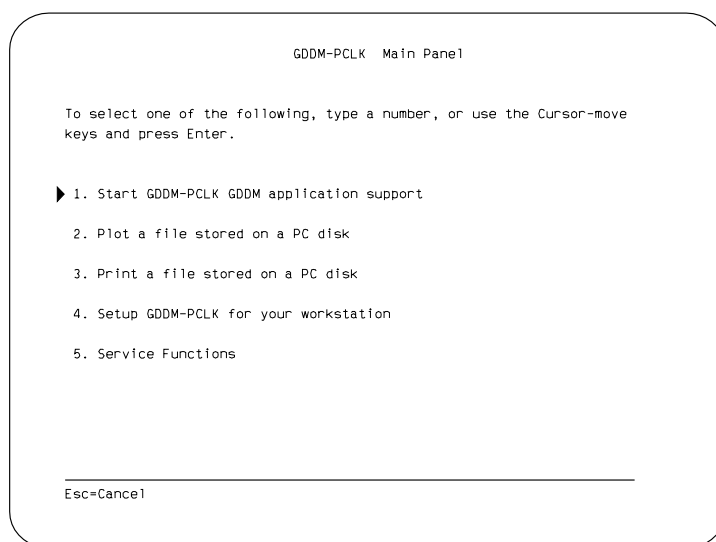


Figure 1. GDDM-PCLK Main Panel

3. Select option 5, Service Functions.

The GDDM-PCLK Service Functions panel is displayed:

```

GDDM-PCLK Service Functions

To select an item, position the cursor against it and press the
Space Bar. Press Enter to save the changes and return to the
Main Panel.

Trace . . . . . Off
                  ▶ On

Output . . . . . File (\pclk11\pclk.trc)
                  Mono Screen
                  ▶ Mono Screen & File

Transmission
Buffer . . . . . ▶ No
                  Yes

Trace Points . . . 01 04 ▶ 07 10 13 ▶ 16 19 22
                   02 05 08 11 ▶ 14 17 20 23
                   03 06 ▶ 09 12 15 18 21 24

Esc=Cancel

```

Figure 2. GDDM-PCLK Service Functions

4. Use this panel to start the GDDM-PCLK trace facility.

If you have a personal computer system with a single screen, you are only given the option of sending the trace output to a file in the \PCLK11 subdirectory. If you have a personal computer system with dual screens, you may be given the options to send the trace output to a file, or to the monochrome screen only, or to both of these.

You can choose to trace the data stream that is sent, through the transmission buffer, to and from the host computer.

Trace points

Trace points 01 through 24 represent the component parts of GDDM-PCLK which you can select to restrict the trace to specific components. The IBM Support Center staff can tell you which trace points to select.

The trace points are:

- 01** XHGI calls
- 02** GQDE – Environmental component
- 03** GQDH – Hardcopy component
- 04** GQDK – Key stroke component
- 05** GQDL – Terminal emulator services component
- 06** GQDP – Panel services component
- 07** GQDS – Supervisor component
- 08** GQDDORD – Drawing order processor
- 09** GQDY – Common services component
- 10** GQDDBPP/GQDDPLT – Built-in procedure/plotter processor
- 11** GQDDCON – Query reply processor
- 12** GQDDFXF/GQDDPRT – File transfer/printer data stream processor
- 13** GQDDGDP/GQDDGDI/GQDDGDR – Graphics data processor
- 14** GQDDODP – Outbound data processor

- 15** GQDI – Installation component
- 16** GQDA – Alphanumeric merge component
- 17** Alphanumeric data
- 18** GQDLKEY – Key-stroke processing
- 19—24** (Spare)

Turning on the trace point of a component activates the tracing of all modules within that component. For example, if trace point 04 is selected, all modules in the Key stroke component are traced.

The data stream component has no global trace point. To trace modules in the data stream component, you have to turn on one or more trace points (08, 10, 11, 12, 13, and 14) depending on the modules you need to trace.

More than one trace point can be active at once if there is enough spare memory to contain the message data for the trace points.

When you have completed the GDDM-PCLK Service Functions panel, press ENTER to save the changes, and then press PF3 to return to the GDDM-PCLK Main Panel. Then perform the operation that was causing the problem.

Here is a trace listing where trace point 01 was chosen:

```

PCLK TRACE                started at 12:30:25 on 02/29/1988

Selected Trace Points are:
TB 01

01 HOPEN
01 XHGI Parameters are:
    03000000 00                *..... *
01 XHGI Reply fs:
    03000000 00                *..... *
01 HQDPS
01 XHGI Parameters are:
    04000000 0020              *..... *
01 XHGI Reply fs:
    04005809 0020              *..X.. *
01 HINIT
01 XHGI Parameters are:
    0200665F                  *..f_ *
01 HQMODE
01 XHGI Parameters are:
    1300665F 00202020 20202020 20475144 *..f_.. GQD*
    58583130 00                *XX10. *

```

Figure 3. GDDM-PCLK Trace

Trace options

When the trace output is sent to the screen, you can choose paged mode or single-line mode. In paged mode, whenever the trace output has filled the screen, GDDM-PCLK beeps and waits for you to press a key. In single-line mode, GDDM-PCLK stops and waits for a key to be pressed after every line of trace output is displayed.

When the trace output is sent to a file, you can choose whether to close and open the trace file for every trace record. If you do this, you do not lose any trace records even if the personal computer system stops working. However, the GDDM-PCLK performance is reduced. The other option is to close and open the

trace file after every 50 trace records. The GDDM-PCLK performance improves, but the latest trace records may be lost if the personal computer system stops working.

At entry to every module you are tracing, you can choose whether to trace the control block and stack dumps.

Trace control keys

You can change the trace options while GDDM-PCLK is in the polling loop, or while it is waiting for a key to be pressed in page mode or single-line mode.

To change an option, press and hold down the Ctrl key and press the other key indicated. Each key combination is a *toggle*. Valid combinations are:

Ctrl+T	Trace
Ctrl+D	Trace to screen
Ctrl+P	Use paged-mode display
Ctrl+S	Use single-line mode display
Ctrl+F	Trace to file
Ctrl+O	Close and open file after each write
Ctrl+B	Trace control block and stack dumps

Trace output

There are four types of trace output:

- Module
- XHGI component
- Transmission buffer
- Control block

Module

When a module is called, the entry to and exit from the module always causes a trace output if tracing is activated for that module. The trace output looks like this:

```
02 INTO GQDEDRV
:
09 INTO GQDYLEV
:
09 LEAVING GQDYLEV - RC=0000
:
02 LEAVING GQDEDRV - RC=0000
```

The indentation of INTO and LEAVING indicates the nesting level of the call. The number on the left-hand side shows to which trace point the trace output belongs.

Many modules give the requested function (why the module was called) and the return code:

```
02 INTO GQDEDRV
02 DRV - Init
:
02 LEAVING GQDEDRV - RC=0000
```

In this example, module GQDEDRV was called for initialization (Init) and ended successfully (RC=0). To find out what module a trace output is from, use the trace

point ID on the left-hand side (for example, 02 is GQDE component), and the last three letters of the module name shown (for example, DRV).

All the other module trace records depend on the module.

The return code from a DOS function call is often traced. Some modules trace the value assigned to an important variable with the variable name. Here is an example:

```
03 PTF - DOSFIRST RC = 00
03 PTF - file_count = 0008
```

A big module often traces the flow of process as follows:

```
07 STC - Initializing ODP
07 STC - Initializing PRT
07 STC - Setting Host Session Number
07 STC - Loading XHGI Driver
07 STC - Initializing GDP
07 STC - Initializing BPP
```

There are other types of trace output depending on the modules; they are mainly self-explanatory.

XHGI component

XHGI is the generic name for a set of program interfaces for display adapters. Calls to the XHGI component from the other components can be traced by activating the trace point 01. The trace output looks like this:

```
01 HQDPS
01 XHGI Parameters are:
      04000000 0020          *.....*
01 XHGI Reply is:
      04005809 0020          *..X..*
```

The trace output of each XHGI call begins with the XHGI entry point name (for example, HQDPS) followed by the parameters passed to the XHGI call both in hexadecimal and ASCII. If the XHGI call returns any parameters to the invoking routine, they are also traced in hexadecimal and ASCII.

The detail of the XHGI parameters can be found in the *IBM Personal System/2 Display Adapter 8514/A Technical Reference*.

Transmission buffer

The following information is traced by selecting the transmission buffer trace in the Service Functions panel:

- Outbound control field
- Outbound data
- Inbound control field
- Inbound data
- Attention identifier (AID) sent

These are usually traced both in nibblized and denibblized form. Each trace record has the trace point ID of "TB" followed by the last three letters of the TES-component module that activated the trace. The time of the trace is output as well.

An example of transmission buffer trace is:

TB RCV - Denibblized Data is :

```

                                18:04:05
F3000403 80000601 FF0380      *..... .. *
```

Control block and stack

If active, the control block and stack are dumped at the entry to every module that is activated through the trace point.

The trace output shows all the GDDM-PCLK control blocks followed by the stack.

The stack is traced from the current top of the stack to the bottom of the stack area. If this is bigger than 512 bytes, only the top 512 bytes of the stack are traced.

Trace message data file

The trace message data file (GQDSTRCA.DAT) contains the text for most of the trace messages. It consists of a header table and a section for each trace point.

When tracing is initialized, the appropriate section of the trace data file is loaded into memory for each trace point selected. When fewer trace points are selected, less memory is required.

Trace code within GDDM-PCLK uses an index into the memory for each trace point.

Changes to the default image symbol set

GDDM-PCLK when installed on a personal computer system, automatically downloads the default image symbol set that it uses to display mode-1 graphics text. For more information about mode-1 graphics, see the *GDDM Base Application Programming Guide*.

You can edit the default image symbol set in the host computer, using the GDDM Image Symbol Editor; see the *GDDM Using the Image Symbol Editor* book. However, the changes are only reflected in GDDM-PCLK if they are made before GDDM-PCLK is installed on the personal computer system. If GDDM-PCLK has already been installed on the personal computer system, the personal computer system user must delete the personal computer system version of the symbol set and run GDDM-PCLK again.

At this stage, the new default image symbol set is automatically downloaded from the host computer and the personal computer system user can see the changes. The personal computer system version of the symbol set is in the \PCLK11 directory on the personal computer system, and has a file name of the format **GQDnXm.SYM**, where *n* and *m* depend on the personal computer system you are using.

Debugging Personal Communications/3270 problems

Personal Communications/3270 is an emulator program for the personal computer system in a DOS environment. If you receive data stream errors during Personal Communications/3270 operations or incorrect output on the associated printer sessions, it is normally easier to trace the process from the emulator program than to trace the host communications using a VTAM trace.

The following section shows how to prepare and format a Personal Communications/3270 trace, and how to collect and prepare the material that the IBM Support Center staff require to diagnose a Personal Communications/3270 problem.

Tracing Personal Communications/3270 Communications

To run the trace, you need to specify the trace option in DOS mode when you load Personal Communications/3270. The following example command loads Personal Communications/3270 programs¹ and provides a 64KB trace table.

```
PC3270 /T=64
```

The trace is written in wrap-around mode. This means that when the trace buffer is full, any more trace data is written into the buffer from its starting address, overwriting the data already there. To capture as much trace data as possible, specify the maximum value of 64KB for the trace buffer size.

Note: If you do not have enough storage available for a 64KB trace buffer, you will receive an error message, but the trace buffer should be at least 10KB in size. You can increase the storage available by removing some Terminate and Stay Resident (TSR) programs before loading Personal Communications/3270.

You should now recreate the error you need to report.

After the error situation is recreated, switch to the DOS session and use the following command to format the in-storage trace to a personal computer system file:

```
PCSMON >drive:filename.extension
```

For example, to format the trace to the file EXAMPLE.TRC on drive C, enter:

```
PCSMON >c:example.trc
```

The current contents of the Personal Communications/3270 trace table is then saved into the personal computer system file.

Note: If you specified a password during the tracing process, it is visible in the trace.

¹ Assuming the start command is PC3270.BAT

Reporting Personal Communications/3270 Problems

You should submit a problem report for Personal Communications/3270 to the IBM Support Center staff. You should also send them copies of the following files:

- The AUTOEXEC.BAT file
- The CONFIG.SYS file
- The complete Personal Communications/3270 subdirectory including:
 - The STARTUP.BAT file
 - The printer definition files
 - The keyboard definition files
- The PCSMON trace file
- A description of your system setup, including:
 - The type of host used (VM/MVS....)
 - The host software used (IND\$FILE, IMS,.....)
 - The type of connection used (DFT/CUT/Token Ring...)
 - The physical connection (control unit type / line type / microcode levels...)
 - A description of the personal computer systems hardware used, including the installed adapter cards and monitors
 - A description of the printer used
- A file describing the circumstances in which the error occurred, including the error messages received.

Chapter 7. GDDM-OS/2 Link diagnosis

If IBM personal computer system users have a problem running GDDM-OS/2 Link, they must report the problem to you.

Note: Personal computer system users *cannot* report the problem directly to the IBM Support Center staff.

Before reporting a problem to the IBM Support Center staff, ensure that it is not a restriction with GDDM-OS/2 Link support by referring to the section “Restrictions with GDDM-OS/2 Link” in the online *GDDM-OS/2 Link User's Guide*. If you cannot solve the problem yourself, call the IBM Support Center, quoting the component identifier for the GDDM Base system on which GDDM-OS/2 Link is installed. For information on which component identifier applies to your system, and for any general points about defining problem symptoms to the IBM Support Center staff, see Chapter 8, “Reporting GDDM problems to IBM” on page 135.

This chapter tells you:

- How service is applied to GDDM-OS/2 Link
- How to find the level of GDDM-OS/2 Link code
- How to use GDDM-OS/2 Link trace support
- How to perform a communications trace under the OS/2 Communications Manager
- How to understand entries made in the OS/2 Communications Manager error log by GDDM-OS/2 Link
- How to complete an OS/2 problem report form

Finding the level of OS/2 code

When reporting a problem, you may need to know the level of OS/2 code installed on a personal computer system. To do this, enter an OS/2 Full Screen Session and issue the following command.

```
SYSLEVEL
```

```
C:\OS2\INSTALL\SYSLEVEL.OS2
IBM OS/2 Base Operating System
Extended Edition 1.30      Component ID 566933601
Current CSD level: WR0500
Prior   CSD level: WR0500

C:\CMLIB\SYSLEVEL.ACS
IBM OS/2 Communications Manager
Extended Edition 1.30      Component ID 566933602
Current CSD level: WR0500
Prior   CSD level: WR0500

D:\SQLLIB\SYSLEVEL.SQL
IBM OS/2 Database Services
Extended Edition 1.30      Component ID 566933603
Current CSD level: WR0500
Prior   CSD level: WR0500

Press Enter (↵) to display next page.

Enter
```

Figure 4. GDDM-OS/2: Example of SYSLEVEL output

Applying service to GDDM-OS/2 Link

Service for GDDM-OS/2 Link is always applied to the GDDM Base program. When personal computer system users start a host GDDM application, they are presented with a menu from which they can choose to download the service files from the host computer to the personal computer system on which GDDM-OS/2 Link is installed.

Finding the level of GDDM-OS/2 Link code

When reporting a problem, you may need to know the level of GDDM-OS/2 Link installed on a personal computer system.

This can be done either by starting the service level utility from the GDDM-OS/2 Link group, or by changing directory to X:\CMLIB in an OS/2 session (where X is the drive on which you installed the OS/2 Communications Manager) and issuing the following command:

```
GQFLEVEL
```

A list of the GDDM-OS/2 Link files and their level numbers is displayed. These level numbers indicate to the IBM Support Center staff what service has been applied.

A typical display is as follows:

```
\CMLIB:
GQFTRACE.EXE      19216449      *
GQFINST.EXE       -----
GQFLEVEL.EXE      19216448      *
GQFNHUS.HLP       19216450      *
GQFFON.FON        -----
GQFCLEAN.EXE      19216449      *
GQFNBUS.INF       19216450      *
```

GQFNYUS.HLP	19216449	*
\CMLIB\DLL:		
GQFLAPIO.DLL	-----	
GQFBPS.DLL	-----	
GQFSTUB.DLL	-----	
GQFSYS.DLL	19216460	*
GQFUPD.DLL	19216449	*
GQFFON.DLL	19216449	*
GQFNLUS.DLL	19216450	*
GQFFILES.DLL	19216449	*
GQFTRC.DLL	19216450	*

Notes:

1. 1xxxxxxx against a file name is the level number.
2. * against a file name means that the file can be serviced by GDDM-OS/2 Link.
3. ----- against a file name means that no level information was found. This probably means that the file is part of the emulator.

A problem exists if a file name has both * and ----- against it.

The displayed information is also written to a file called GQFLEVEL.DAT.

GDDM-OS/2 Link trace support

For some problems, the IBM Support Center staff may ask you to investigate further. If so, you may need to run a trace.

For GDDM-OS/2 Link, two types of trace support are available:

- The GDDM-OS/2 Link Service Trace Utility (GQFTRACE.EXE)
- The trace component of the OS/2 Communications Manager

Note: You may also need to run a GDDM trace in conjunction with the GDDM-OS/2 Link or OS/2 traces. See Chapter 2, “GDDM tracing” on page 21 for details on how to do this.

Running the GDDM-OS/2 Link Service Trace Utility

Before starting the trace utility, you must stop all 3270 terminal emulation sessions. Then start the trace utility, either from the GDDM-OS/2 Link group, or by issuing the following command from an OS/2 session:

```
GQFTRACE
```

The trace utility is started minimized with tracing to file only selected (see the Output option in the section “Options Menu” on page 111.) As each traceable process of GDDM-OS/2 Link starts, default trace points are automatically selected for tracing.

After you start the trace utility, restart the 3270 terminal emulation sessions.

The default trace output file name of C:\GQFTRACE.DAT can be changed by specifying a different name as a parameter to GQFTRACE. You can do this in one of two ways:

- By entering the following command from an OS/2 session:
GQFTRACE A:\CMLIB\PROBLEM.TRC
- By changing the settings for the trace utility in the GDDM-OS/2 Link group.

The current trace output name is displayed on the GDDM-OS/2 Link Service Trace Utility title bar.

The trace utility can produce several error messages. These are:

GQF0900	Initialization error: Insufficient memory.
GQF0910	Initialization error: Program is already running.
GQF0920	Initialization error: Program was unable to start.
GQF0930	Initialization error: Program is unable to start. A program previously traced is still running.
GQF0940	File error: Unable to open output file file name. Tracing to file is disabled.
GQF0950	File error: DOS file error; reason code = error number. Tracing to file is disabled.
GQF0960	Help error: Unable to load help; reason code = error number. Help is disabled.

Note: Before stopping the GDDM-OS/2 Link trace utility, you must stop all 3270 terminal emulation sessions.

GDDM-OS/2 Link Service Trace Utility pull-down menus

The following pull-down menus, which appear on the GDDM-OS/2 Link Service Trace Utility menu bar, enable you to change the way the GDDM-OS/2 Link Service Trace Utility works or provide help information.

Note: Contextual help is available from all pull-down menus.

File Menu

Item	Description
Marker	This determines the type of marker that appears in the trace output. It can be set to be a timestamp or can be a user-defined character string up to a maximum of 80 characters.
Exit	This exits the trace program.

Trace Options Menu

This menu allows you to select different trace points to trace for each process. It is initially grayed-out until the first process to be traced is started.

If you select any of the options from this menu, one of the windows in “Trace options windows” on page 112 is displayed.

Item	Description
Update	This displays Figure 5 on page 112, the Update Trace Options window.
Environment	This displays Figure 6 on page 113, the Environment Trace Options window.
Window Procedure	This displays Figure 7 on page 114, the Window Procedure Trace Options window.
Session n	This displays Figure 8 on page 115, the Session n Trace Options window, where n is the short session identifier.

See “Trace options windows” on page 112 for displays of these windows and descriptions of the trace points which you can select from each one. Where the phrase “not used” appears grayed-out against any of the trace points, those trace points are not applicable to that particular window.

Trace points 01 through 16 on these windows represent the different component parts of the processes, so that you can restrict the trace to specific parts by making one or more selections. Turning on the trace point of a component activates the tracing of all modules within that component.

The IBM Support Center staff can tell you which trace points to select.

Options Menu

Item	Description
Output	<p>This determines whether the trace output appears on the screen, or is written to a file, or both.</p> <p>If the trace output is written to a file and no output file has been specified, a default output file GQFTRACE.DAT is created in the root directory of the C: drive.</p>
Autosave	<p>This determines how often trace information is written to disk when tracing to file is active.</p> <p>If you close and open the trace file for every trace record, you do not lose any trace records even if the personal computer system stops working. However, the GDDM-OS/2 Link performance is reduced.</p> <p>If you close and open the trace file after every 50 trace records, the GDDM-OS/2 Link performance improves, but the most recent trace records may be lost if the personal computer system stops working.</p>
Font	This determines the size of the font (small/large) when tracing to screen is active.

Slow This slows down the trace program, which may be necessary to recreate certain problem situations.

Help Menu

Item	Description
Help for Help	This provides the standard help overview panel.
Extended Help	This provides general information about the trace program.
Help Index	This provides an index of all the entries in the help file.

Trace options windows

Note: The windows shown here are typical; they may vary in size and format according to the type of display in use.

Update

Figure 5. GDDM-OS/2 Link: Update Trace Options window

Trace point 15: Update

Used by the service update component.

Environment

[02] GDDM-OS/2 Link Environment	
<input type="checkbox"/> [01] n/a	<input checked="" type="checkbox"/> [09] H Host internal
<input checked="" type="checkbox"/> [02] D Dialog/Hsg	<input type="checkbox"/> [10] n/a
<input checked="" type="checkbox"/> [03] E Envr in/out	<input checked="" type="checkbox"/> [11] Q PH Query
<input checked="" type="checkbox"/> [04] E Envr internal	<input type="checkbox"/> [12] n/a
<input type="checkbox"/> [05] n/a	<input type="checkbox"/> [13] n/a
<input type="checkbox"/> [06] n/a	<input checked="" type="checkbox"/> [14] H Tx/Rx data
<input type="checkbox"/> [07] n/a	<input checked="" type="checkbox"/> [15] Z Utilities
<input checked="" type="checkbox"/> [08] H Host in/out	<input type="checkbox"/> [16] n/a
<input type="checkbox"/> Show all highlight messages	
<input type="button" value="Enter"/> <input type="button" value="Cancel"/> <input type="button" value="All set"/> <input type="button" value="All reset"/> <input type="button" value="Defaults"/>	

Figure 6. GDDM-OS/2 Link: Environment Trace Options window

Trace point 02: D Dialog/Msg

Used by the window, message, and help components.

Trace point 03: E Envr in/out

Used by the high-level environment component to trace entry and exit of each function.

Trace point 04: E Envr internal

Used by the high-level environment component to trace internal processing.

Trace point 08: H Host in/out

Used by the host communication component to trace entry and exit from each function.

Trace point 09: H Host internal

Used by the host communication component to trace internal processing.

Trace point 11: Q PM Query

Used by the query component to trace Presentation Manager Query processing.

Trace point 14: H Tx/Rx data

Used to trace the transmission buffer data sent and received from the host.

Trace point 15: Z Utilities

Used by the general-purpose utilities component.

Window procedure

[02] GDDM-OS/2 Link Window procedure

<input checked="" type="checkbox"/> [01] B BP1	<input checked="" type="checkbox"/> [09] H Host internal
<input checked="" type="checkbox"/> [02] D Dialog/Hsg	<input checked="" type="checkbox"/> [10] Q D/S query
<input checked="" type="checkbox"/> [03] E Envr in/out	<input checked="" type="checkbox"/> [11] Q PM query
<input checked="" type="checkbox"/> [04] E Envr internal	<input checked="" type="checkbox"/> [12] O ODP
<input checked="" type="checkbox"/> [05] F File Xfer	<input checked="" type="checkbox"/> [13] W Winproc
<input checked="" type="checkbox"/> [06] G GOCA flow	<input checked="" type="checkbox"/> [14] H Tx/Rx data
<input checked="" type="checkbox"/> [07] G GOCA data	<input checked="" type="checkbox"/> [15] Z Utilities
<input checked="" type="checkbox"/> [08] H Host in/out	<input type="checkbox"/> [16] n/a

☐ Show all highlight messages

Enter Cancel All set All reset Defaults

Figure 7. GDDM-OS/2 Link: Window Procedure Trace Options window

Trace point 02: D Dialog/Msg

Used by the window, message, and help components.

Trace point 03: E Envr in/out

Used by the high-level environment component to trace entry and exit of each function.

Trace point 04: E Envr internal

Used by the high-level environment component to trace internal processing.

Trace point 13: W WinProc

Used by the GDDM-OS/2 Link window procedure.

Trace point 15: Z Utilities

Used by the general-purpose utilities component.

Session n

[On] GDDM-OS/2 ODP LT Name N

<input checked="" type="checkbox"/> [01] B BP1	<input checked="" type="checkbox"/> [09] H Host internal
<input checked="" type="checkbox"/> [02] D Dialog/Hsg	<input checked="" type="checkbox"/> [10] Q D/S query
<input checked="" type="checkbox"/> [03] E Envr in/out	<input checked="" type="checkbox"/> [11] Q PH Query
<input checked="" type="checkbox"/> [04] E Envr internal	<input checked="" type="checkbox"/> [12] O ODP
<input checked="" type="checkbox"/> [05] F File Xfer	<input type="checkbox"/> [13] n/a
<input checked="" type="checkbox"/> [06] G GOCA flow	<input checked="" type="checkbox"/> [14] H Tx/Rx data
<input checked="" type="checkbox"/> [07] G GOCA data	<input checked="" type="checkbox"/> [15] Z Utilities
<input checked="" type="checkbox"/> [08] H Host in/out	<input type="checkbox"/> [16] n/a

☐ Show all highlight messages

Enter Cancel All set All reset Defaults

Figure 8. GDDM-OS/2 Link: Session n Trace Options window

Note: In the above window heading, the identifier [03] and the LT name A vary according to which host session logical terminal (LT) the trace process is for.

Trace point 01: B BP1

Used by the graphic input device component.

Trace point 02: D Dialog/Msg

Used by the window, message, and help components.

Trace point 03: E Envr in/out

Used by the high-level environment component to trace entry and exit of each function.

Trace point 04: E Envr internal

Used by the high-level environment component to trace internal processing.

Trace point 05: F File Xfer

Used by the file transfer component when service updates are applied, or when picture interchange files are transferred from the host.

Trace point 06: G GOCA flow

Used by the graphics data stream processing component to trace the flow of control.

Trace point 07: G GOCA data

Used by the graphics data stream processing component to trace the data being processed.

Trace point 08: H Host in/out

Used by the host communication component to trace entry and exit from each function.

Trace point 09: H Host internal

Used by the host communication component to trace internal processing.

Trace point 10: Q D/S query

Used by the query component to trace internal processing.

Trace point 11: Q PM Query

Used by the query component to trace Presentation Manager query processing.

Trace point 12: ODP

Used by the outbound data processing component.

Trace point 14: H Tx/Rx data

Used to trace the transmission buffer data sent and received from the host.

Trace point 15: Z Utilities

Used by the general-purpose utilities component.

Tracing output

When a module is called, the entry and exit from the module always cause trace output if tracing is activated for that module. The return code is also output. A typical trace output looks like this:

```
[01] (03) INTO GqfeInitFontFile
[01] (04) EIFON:Font input name =
           433A5C4F 53325C44 4C4C5C44 4953504C
           41592E44 4C4C
[01] (03) INTO GqfeGetPathNames
[01] (04) Filename is:
           433A5C43 4D4C4942 5C444C4C
[01] (04) Filename is:
           433A5C43 4D4C4942 5C
[01] (03) LEAVING GqfeGetPathNames rc=00000000
[01] (04) EIFON:Font base name =
           433A5C43 4D4C4942 5C444C4C 5C475146
           464F4E2E 444C4C
[01] (04) EIFON:Font output name =
           433A5C43 4D4C4942 5C475146 464F4E2E
           464F4E
[01] (15) INTO GqfzMakeFontFile
[01] (15) ZMFON:Input file handle = 00000000
[01] (15) ZMFON:Input file date = 173C, time = 7397
[01] (15) ZMFON:Input file size = 000454B0
[01] (15) DosOpen C:\CMLIB\GQFFON.FON, rc=0
[01] (15) ZMFON:Output file already exists
[01] (15) ZMFON:Input file is unchanged, leaving
[01] (15) LEAVING GqfzMakeFontFile rc=00000000
[01] (03) LEAVING GqfeInitFontFile rc=00000000
[01] (04) Saved GQF 3270 HostGraphics,GQF Common in OS2.INI
[01] (03) LEAVING GqfeInitFontFile rc=00000000
[01] (04) ECTL:Prime anchor 04E70008
[01] (04) INTO GqfeGetTraceInfo
[01] (04) EGTRC:Lancia Shared Seg addr = 1B170000
[01] (04) EGTRC:trace flags addr=1B170553, setting=0
[01] (04) EGTRC:comp name = HOSTGRAF
[01] (04) LEAVING GqfeGetTraceInfo rc=00000000
```

The number in square brackets on the left-hand side, for example [01], shows the process to which the trace output belongs. This number also appears on the title bar of the corresponding trace options window. For example [01] appears on the title bar of Figure 6 on page 113.

The indentation between INTO and LEAVING indicates the nesting level of the call. The number in parentheses on the left-hand side, for example (08), shows the trace point to which the trace output belongs.

Some modules also trace the value assigned to an important variable. Here is an example:

```
[01] (15) ZMFON:Input file date = 173C, time = 7397
```

There are other types of trace output depending on the modules. They are mainly self-explanatory.

Trace as part of the OS/2 Communications Manager

GDDM-OS/2 Link uses the OS/2 Communications Manager Common Services for trace. This assists problem isolation among the different products. The Common Services use the same test points as GDDM-OS/2 Link, but put the output in a trace buffer. There are two trace controls: code 27 for data stream, and code 28 for the parameters and communication flow.

Sequential tracing, when data flows from one device to another, can be done through this trace. For information about how to run this trace, see the OS/2 Communications Manager problem determination documentation.

Performing a communications trace

This section gives procedures for obtaining trace data during normal communications operation and during an OS/2 Communications Manager auto-start.

Obtaining trace data during normal communications operation

Follow the steps below to perform a normal communications trace:

1. To begin:
 - a. Stop and restart the OS/2 Communications Manager.
 - b. Record the date and time at which the system is started so that the error and message log entries associated with the re-creation of the problem can be identified.
2. Access the Problem Determination Aids menu:
 - a. Select Advanced from the action bar at the top of the OS/2 Communications Manager Main Menu.
 - b. When the Advanced panel appears, either type 3, or press the P key for Problem determination aids.
3. Choose the events that you need to trace:
 - a. When the Problem Determination Aids menu is displayed, access the Trace Services menu either by typing 2, or by pressing the T key for Trace services.
 - b. When the Trace Services menu is displayed, select the Select traces option.

- c. When the Trace Type Selection panel is displayed, select Trace selection. The IBM Support Center staff may ask you to select Advanced trace selections at this point to get more detailed trace information for them.
 - d. The Trace Selections menu is displayed.
 - e. Make selections as required by using the space bar. Then select Enter. The Trace Services menu is displayed.
4. Start traces:
 - a. From the Trace Services menu, either type 2 or press the T key to start selected traces.
 - b. When the message
Traces have been started.
is displayed, select Esc=Cancel twice to return to the OS/2 Communications Manager Main Menu.
5. Recreate the problem.
6. Save the data trace:
 - a. Access the Trace Services menu again.
 - b. Either type 3 or press the P key to stop traces.
 - c. Copy the storage trace to the file by typing 4 or by pressing the C key. The Copy Storage Trace to File panel is displayed.
 - d. Specify a file name for the file that is to contain the trace data, and select Enter.
 - e. When the copy completes, select Esc=Cancel twice to return to the OS/2 Communications Manager Main Menu.

Obtaining trace data during the OS/2 Communications Manager auto-start

Follow the steps below to perform an auto-start communications trace:

1. To begin:
 - a. Stop and restart the OS/2 Communications Manager.
 - b. Record the date and time at which the system is started so that the error and message log entries associated with the re-creation of the problem can be identified.
2. Access the Problem Determination Aids menu:
 - a. Select Advanced from the action bar at the top of the OS/2 Communications Manager Main Menu.
 - b. When the Advanced panel appears, either type 3, or press the P key for Problem determination aids.
3. Choose the events that you wish to trace:
 - a. When the Problem Determination Aids menu is displayed, access the Trace Services menu either by typing 2 or by pressing the T key for Trace services.
 - b. When the Trace Services menu is displayed, select Auto-trace services by either typing 6 or by pressing the A key.

- c. When the Auto-trace Services panel is displayed, select Select and store auto-traces.
 - d. When the Trace Type Selection panel is displayed, select Trace selection. The IBM Support Center staff may ask you to select Advanced trace selections at this point to get more detailed trace information for them.
 - e. The Trace Selections menu is displayed. Make selections as required by pressing the space bar. Then select Enter.
 - f. The Auto-trace Services panel is displayed.
4. Enable traces:
 - a. Either type 2 or press the E key to enable the selected traces.
 - b. Select Esc=Cancel to return to the Trace Services menu.
 - c. Select Esc=Cancel twice to return to the OS/2 Communications Manager Main Menu.
 5. Stop and then restart the OS/2 Communications Manager.
 6. Save the data trace:
 - a. Access the Trace Services menu (as directed previously).
 - b. Either type 3 or press the P key to stop traces.
 - c. Copy the storage trace to the file by typing 4 or by pressing the C key. The Copy Storage Trace to File panel is displayed.
 - d. Specify a file name for the file that is to contain the trace data, and select Enter.
 - e. When the copy completes, select Esc=Cancel twice to return to the OS/2 Communications Manager Main Menu.

If the OS/2 Communications Manager does not initialize successfully, you will not be able to access the Problem Determination Aids menu when you reach step 2 on page 118. If this occurs, you may be able to gather trace data by following the alternative procedure given next:

1. Stop and restart the OS/2 Communications Manager with a configuration file that allows the OS/2 Communications Manager to initialize successfully.
2. From the OS/2 Communications Manager Main Menu, press the Switch to Action Bar (F10) key to go to the action bar.
3. Select Advanced from the Main Menu action bar. The Advanced Options panel is displayed.
4. Select Configuration on the Advanced Options panel. The Specify Configuration File panel is displayed. Specify the configuration file with which the OS/2 Communications Manager is failing to initialize and select Enter.
5. The Communications Configuration Menu is displayed. Select Workstation Profile from this panel. The Profile Operations panel is displayed. Select Change from this panel.
6. The Change Workstation Profile panel is displayed. Select Enter without modifying anything. The configuration file is now identified as modified and unverified. Exit Configuration without verifying the configuration file.

7. Stop and restart the OS/2 Communications Manager with the unverified configuration file. The OS/2 Communications Manager should initialize successfully.
8. Follow steps 2 through 4 of the previous procedure to select and enable auto-traces in the unverified configuration file and to return to the OS/2 Communications Manager Main Menu.
9. Follow steps 2 through 4 of this procedure to get to the Communications Configuration Menu. Verify the Configuration file that has the auto-traces enabled. Exit Configuration.
10. Stop and restart the OS/2 Communications Manager with the configuration file that now has the auto-traces enabled and that was verified in step 9. The OS/2 Communications Manager again fails to initialize, but the trace data is automatically written to a file named ACSTRACE.DAT. The trace data can be printed or viewed by using a standard print command or any editor.

Using the Error Log entries from the OS/2 Communications Manager

If GDDM-OS/2 Link detects an error, it enters the following details into the OS/2 Communications Manager error log:

- Type
- Subtype
- Originator
- Conversation ID
- Process ID
- Error Data

The error data field contains this error-dependent information:

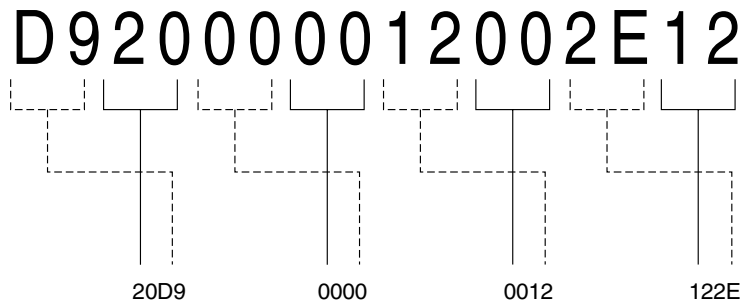


Figure 10. Example of log message error data

For example, if the error data is D920 (see Figure 10 above), this represents error code 20D9, which is PMERR_METAFILE_IN_USE.

For error log entries with subtype = 0000002, the error data field contains a DOS error code. The error code is the first two bytes, which must be byte-reversed for correct interpretation. DOS errors can be identified using the OS/2 HELP command, by converting the error code from hexadecimal to decimal. For example, if the error data is 1500, this represents error code 0015, which is 21 in decimal. So you must type `HELP 21`.

For error log entries with other subtype values, the error data field contains information on an error generated by GDDM-OS/2 Link, which should be specified if you report the error to IBM. See Chapter 8, "Reporting GDDM problems to IBM" on page 135 for details. IBM Support Center staff can find more detailed information in the appropriate service documentation.

How to complete an OS/2 Problem Report Form

This part of the chapter contains the OS/2 program Problem Report Form (PRF), describes when and why a PRF should be used, and explains how to complete a PRF.

Do not write on the blank Problem Report Form. Make several copies and save this original in case you need additional copies in the future.

When and why you should use this form

IBM suggests that service coordinators maintain a history of problems that they have been asked to investigate. Make several copies of the blank Problem Report Form (PRF) contained in this part for future use.

When a problem is reported, use a copy of the PRF to maintain a history and to report problems to IBM.

How to use a Problem Report Form

Answer all the questions in Part A and Part B, and, if applicable, the questions in Part C for the communications problems only. Explanatory information for each question is provided as follows.

Part A: BACKGROUND INFORMATION:

1.a. Customer Number: This number is assigned at the time the program license is acquired. If you do not know your customer number, contact your authorized IBM marketing representative.

1.b. Customer Name: Company or organization.

1.c. Your Name

2. Problem Numbers: If the problem was previously reported to the IBM Support Center, one or both of the following numbers may have been assigned by IBM to the problem:

- Problem Management Record Number.

This number can be provided from the IBM Support Center representative when you report a problem, if a resolution was not provided at the time of your call.

- Authorized Program Analysis Report (APAR) Number.

If the problem is new and unique, IBM will continue to research potential causes and actions. For purposes of IBM internal recording, this process creates an IBM report known as an Authorized Program Analysis Report (APAR).

If either or both numbers are assigned to the problem, write the numbers in the spaces provided on the PRF (Part A, question 2).

3.a. Program Name: This is filled in for you.

3.b. Version Number: The release of the product.

3.c. Level: If you have previously loaded program updates or corrections from IBM, write the current level in the space provided. Use SYSLEVEL to determine the current level. See "Finding the level of OS/2 code" on page 107.

3.d.Extended Edition (EE) or Extended Services (ES) Place a checkmark (X) against the relevant component.

3.e.EE or ES level If you have previously loaded program updates or corrections from IBM, write the current level in the space provided.

3.f.Output from OS/2 Link GQFLEVEL program Include the filenames and the level numbers reported by the GQFLEVEL program.

A typical display is as follows:

```
\CMLIB:
  GQFTRACE.EXE      19216449      *
  GQFINST.EXE       -----
  GQFLEVEL.EXE      19216448      *
```

Notes:

1. 1xxxxxxx against a file name is the level number.
2. * against a file name means that the file is serviceable by GDDM-OS/2 Link.
3. ----- against a file name means that no level information was found. This probably means that the file is part of the emulator.

A problem exists if a file name has both * and ----- against it.

This information is also written to a file called GQFLEVEL.DAT.

3.g. Component reporting the problem: Check (X) the component that is reporting the problem, such as base operating system.

4. Priority: Read the Priority Checklist that follows and on the Problem Report Form, place a check mark (X) next to the number below that best describes the impact of this problem.

5. Problem Abstract: This abstract should be constructed based upon your responses to other questions on the PRF.

An IBM Support Center representative may use the information in this abstract as the basis of a search through the Known Problem Database. To expedite and simplify this search, copy the following items into the space provided on the PRF for Part A, question 5 (each item should be separated by a blank space):

- The abbreviation for the problem symptoms that you checked in Part B, question 1.

For example, if you checked ABEND for abnormal termination, write ABEND in the space provided in "Problem Abstract."

- For problems involving messages, refer to "MSG for Error Message" under Part B, question 1 on the completed PRF and copy the following information for the problem abstract:

- The abbreviation for message: MSG
- The message identification number
- Any return codes or error codes

- The originator or environment (if one is provided, use the abbreviation; for example, write RAS rather than Problem Determination).

These items are explained in the discussion of Part B that follows.

- The answers to the questions you completed in Part D

For example, for Part D, questions 3, 4 and 5, you might have written CICS, SDLC, and Hayes (for modem type), respectively. In this example, you would write each of these abbreviations or words in the line for "Problem Abstract" following the abbreviation for the problem symptom (separating each entry with a blank space).

- Other pertinent highlights (as space allows) from Part B, question 2.

For example, you might include the name of the last command typed or the last completed task before the problem occurred.

Prioritizing the problem: Having noted the symptoms and special conditions that can be associated with the user's problem, you should estimate how the problem impacts the user's operations.

The following are four sets of descriptions of the impacts that this problem might have on the user's operations. Determine which set best describes the impact of the problem. Then mark Part A, question 4 on the PRF as directed.

Note: If you prefer to write on the Priority Checklist or place a check mark (X) next to the appropriate description, you are recommended to make at least one copy of the Priority Checklist and save the original for future use.

- If the following descriptions apply to this problem:

- The problem affects the use of the OS/2 program but does not severely limit the user's ability to complete any task.
- Despite the problem, the user can complete desired tasks in a way that does not require more time, too many steps, or is annoying to the user.

If these descriptions apply, place a check mark (X) next to Priority 4 for Part A, question 4 on the PRF.

- If the following descriptions apply to this problem:

- The problem is an annoyance to the user due to differences between the design (the way it works) or terminology associated with the OS/2 program and previous or other systems the user is familiar with.
- The problem causes the user to perform unnecessary steps.
- The problem results from a lack of clear, accurate, or complete documentation by IBM, but is nevertheless a problem from which the user can recover.
- The user can complete desired tasks in spite of the problem but the required effort requires more time, too many steps, or is annoying to the user.

If these descriptions apply, place a check mark (X) next to Priority 3 for Part A, question 4 on the PRF.

- If the following descriptions apply to this problem:

- The problem does not occur frequently but is difficult to recover from when it does occur.
- The problem causes frustration for several users but does not prevent them from completing any job-related tasks.
- The problem significantly reduces the productivity of users.
- The problem causes difficulties for other system users (in addition to the user who reported the problem) or produces a conflict of interests among multiple users of the same system or network.

If these descriptions apply, place a check mark (X) next to Priority 2 for Part A, question 4 on the PRF.

Figure 11. Priority Checklist (Part 1)

- If the following descriptions apply to this problem:
 - The problem causes the user to lose data that has been created or is being used.
 - This a frequent problem that is difficult to recover from.
 - The problem creates a situation that is or may be potentially detrimental to the user's safety.
 - The problem prevents the completion of a task the user is normally capable of, or allows the user to complete the task only after a period of time that is longer than average for this task and is causing a loss of productivity.
 - The problem results in inadvertent damage to the system during the completion of a supported operation or problem determination procedure, or while recovering from an exception or error condition.

If these descriptions apply, place a check mark (X) next to Priority 1 for Part A, question 4 on the PRF.

Figure 12. Priority Checklist (Part 2)

Part B: PROBLEM DESCRIPTION: If you have not completed this section based upon directions provided elsewhere in this guide, complete the following as directed.

1. Problem Symptoms: If a message that is somehow associated with the problem that is displayed on a user's screen or was sent to a message log, and if you have not already done so, place a check mark (X) next to MSG and ask the user to provide:

- The 8-digit alphanumeric message identification number
- Any return or error codes contained in the message (if applicable)
- The complete message text

2. Describe the problem.

3. List any traces and dumps.

Part C: WORKSTATION HARDWARE CONFIGURATION INFORMATION:

Provide a description of the type of hardware listed in Part C for the workstation where the problem occurred or was reported. Include the model type (IBM AT or PS/2 models) and model number along with the other information requested, if applicable. Be sure to include the name of the manufacturer and a part number for all listed pieces of hardware.

Note: For LAN-related problems, attach a similar description for each workstation that is associated with the problem.

Part D: HOST COMPUTER AND NETWORK INFORMATION: The questions in this section of the form apply for problems associated only with communications, including any problems experienced while using or reported by Communications Manager.

If necessary (and possible), contact your host personnel to determine the information requested.

Part E: LAN REQUESTER-SERVER INFORMATION: Complete this section if the user was using the OS/2 LAN Requester or OS/2 LAN Server at the time the error occurred.

Provide this information for each workstation that is associated with the problem.

Part F: PROBLEM RESOLUTION: Describe the resolution of the problem in the space provided; if necessary, attach additional text.

Problem Report Form

Part A: BACKGROUND INFORMATION

- 1.a. Customer Number: _____ 1.b. Customer Name _____
- 1.c. Service Coordinator Name: _____
2. Problem Numbers:
 Problem Management Record Number: _____
 Authorized Program Analysis Report (APAR) Number: _____
- 3.a. Program Name: OS/2 Extended Edition or
 OS/2 Standard Edition
- b. OS/2 Version Number: 1.3, 2.0
- c. Level: _____
- d. Extended Edition or Extended Services?: EE () ES ()
- e. EE or ES Level: _____
- f. Output from OS/2 Links GQFLEVEL program: _____
- g. Component reporting the problem: place a check mark (X) next
 to one or more:
- | | |
|--|---|
| <input type="checkbox"/> Base Operating System | <input type="checkbox"/> Database Manager |
| <input type="checkbox"/> LAN Requester* | <input type="checkbox"/> Communications Manager |
- *For LAN problems, also complete Part E: LAN Requester-Server
 Information
4. Priority.
 Check the appropriate priority:
- | | | | |
|----------------------------|----------------------------|----------------------------|----------------------------|
| Priority 1 | Priority 2 | Priority 3 | Priority 4 |
| <input type="checkbox"/> 0 | <input type="checkbox"/> 0 | <input type="checkbox"/> 0 | <input type="checkbox"/> 0 |
5. Problem Abstract: _____

Part B: PROBLEM DESCRIPTION

1. Check one or more problem symptoms:

☐ INSTALL Installation

☐ ABEND Abnormal Termination

☐ WAIT/LOOP Wait or Loop

☐ DOC User Documentation

☐ PERFM Performance Problem

☐ INCORROUT Incorrect Output

☐ MSG Error Message

(for messages, provide the following information if appropriate)

Message Identification Number:

Return Codes and Error Codes:

Originator or Environment (if applicable):

Message Text (optional--see related directions for Part B):

2. Describe the problem in the following space:

3. List any traces and dumps that you have completed:

Part C: WORKSTATION HARDWARE CONFIGURATION INFORMATION

1. What is the computer type and model of the involved workstation?

2. What was the IBM plant of manufacture (if known) and serial number for this computer?

3. How much memory is installed on the workstation?

4. For fixed-disk systems, what is the size of the fixed disk?

5. List any adapters installed in the workstation in the following space (include the manufacturer and part number):

**Part D: HOST COMPUTER AND NETWORK INFORMATION COMPLETE
THIS SECTION FOR COMMUNICATIONS PROBLEMS**

1. What is the host computer operating system?
2. What is the release level of the host system?
3. What is the host application (CICS, TSO, VM, or CMS)?
4. How is the personal computer connected to the host network?
5. If SDLC or ASYNC, what is the modem type, make, and line speed?
6. If ASYNC protocol converter, what is the type, and model number?
7. If DFT or Token-Ring, what is the controller model and level?
8. For file transfer problem, what is the IND\$FILE
version number of the host file transfer program?
9. If X.25, what is the name and type of the network?

PART E: LAN REQUESTER-SERVER INFORMATION

1.a. If this is a server, provide the following information:

OS/2 LAN Server	Version number:	Level:
-----------------	-----------------	--------

1.b. If this is a DOS requester or an OS/2 LAN Server that supports remote program load, provide the following information, if applicable:

IBM DOS	Version number:	Level:
---------	-----------------	--------

IBM PC LAN Program	Version number:	Level:
--------------------	-----------------	--------

IBM LAN Support Program	Version number:	Level:
-------------------------	-----------------	--------

IBM PC Network Protocol	Version number:	Level:
-------------------------	-----------------	--------

3. List the LAN services that were running on this machine when the problem occurred:

4. List the name and version number of any tested and compatible software involved prior to or during the time the problem occurred. Include if the software requires configuration data.

Name of Software	Version Number	Configuration Data
------------------	----------------	--------------------

Part F: PROBLEM RESOLUTION

1. Describe the resolution of the problem in the following space.
If necessary, attach additional text.

Chapter 8. Reporting GDDM problems to IBM

If you find a problem in one of the licensed programs covered by this book, you should report the problem to the IBM Support Center staff. The easiest way to do this is by using one of the electronic network tools listed on page 135. If you do not have access to any of these tools, you must fill in a problem report form (see page 137 for the standard problem form, or Chapter 7, “GDDM-OS/2 Link diagnosis” on page 107 for an OS/2 report form if the problem is with OS/2), and pass the form to the Support Center. The Support Center staff will search the Software Support Facility (SSF) database to determine whether your problem has already been reported, and whether a fix is available.

If you do have to report a problem to the IBM Support Center, quote the component identifier (component ID) for GDDM. This number depends on which version of the GDDM Base licensed program you are using on your host computer. The versions and corresponding component identifiers, program numbers, and FMID, CLC, or release level are given in the section called “Program Support” in the:

- *GDDM/MVS Program Directory*
- *GDDM/VM Program Directory*
- *GDDM/VSE Program Directory*

Quoting the component ID helps the IBM Support Center staff to check whether the problem is known, and if so, which fix to supply for your host computer configuration.

If the problem is not already known, the IBM Support Center staff may ask you to submit an Authorized Program Analysis Report (APAR). For more information, see “Submitting an APAR” on page 142.

For some problems, the IBM Support Center staff may ask you to investigate the problem further. You may need to run a GDDM trace or use the diagnostic facility of GDDM-IMD, GDDM-REXX, GDDM-PCLK, or GDDM-OS/2 Link. These aids are described earlier in this book.

All problems concerning GDDM-PCLK and GDDM-OS/2 Link are reported through the normal GDDM support centers.

Note: Only system support personnel who support the GDDM Base program on your host computer can be the focal point for any contact with the IBM Support Center staff.

Reporting a problem using a network tool

You may report a problem directly to the IBM Support Center staff by using any of the following electronic network tools:

- Information/Access
- IBMLink
- DIALIBM
- SoftwareXcel Extended

Reporting a problem using a problem report form

Note: There is a separate OS/2 report form in Chapter 7, “GDDM-OS/2 Link diagnosis” on page 107 for OS/2 problems.

Complete a problem report form whenever you need to define a GDDM problem, even if the information you record is not to be passed on later to the IBM Support Center staff. You can use completed forms to create and maintain a local register of problems and solutions. Such a register could help you solve some problems without the need to call the IBM Support Center staff.

If you need further guidance about the information you must supply, see “Defining the symptom of the problem” on page 138, “Defining the operating environment” on page 140, or “Defining the devices in use” on page 141.

You are recommended to prepare several copies of the master problem report form (on page 137), and distribute those copies among the system support and administrative personnel at your enterprise.

Do not write on the master problem report form.

IBM PROBLEM REPORT FORM

1. GDDM version, release, and modification: _____
2. GDDM Program Update Tape (PUT) level: _____
3. Operating system/subsystem: _____
4. Operating system/subsystem
release and PUT levels: _____
5. Controller type and model: _____
6. Controller configuration
and code level: _____
7. Device type and model:
(including printers): _____
8. If VTAM or VSCS, show PSERVIC: _____
9. Device tokens in use: _____
10. If CICS, DFHPCT SCRNSZE, DFHTCT feature: _____
11. If ABEND, give ABEND code, CSECT name, and offset: _____

12. Brief description of the problem including full message text:

13. Applicable entries in ADMDEFS: _____

14. If ABEND, give registers and PSW: _____

15. If a trace is available, state type of trace: _____

Defining the symptom of the problem

The symptom is whatever you first notice that leads you to think that there is a problem. Symptoms can be grouped into the following categories, each of which corresponds to an APAR type:

- Unexpected, incorrect, or missing output
- Device program check
- GDDM message
- GDDMabend
- Loop
- Reduced performance
- Wait
- Error in the GDDM documentation

The following sections of this chapter describe these categories in more detail. Select the category that best describes your problem.

Unexpected, incorrect, or missing output

This is APAR type 'INCORROUT'.

This includes output to displays, printers, plotters, and GDDM files (such as GDF files). When you report the problem, describe what output you expected and what you received that led you to suspect an error.

Note: Unexpected output from the GDDM/TSO Print Utility may have some GDDM messages associated with it. These appear on the system console log.

Device program check

This is APAR type 'PROGCKxxx'.

This is an error detected by a terminal, printer, or plotter. On a terminal, the error usually appears as 'PROGxxx' in the operator information area. On a printer, a code may be displayed on a panel at the front of the device. Further device-specific problem analysis may be needed; the IBM Support Center staff can advise you.

GDDM message

This is APAR type 'MSGxxxxxxx'.

Some GDDM messages contain variable fields, for example:

```
ADM0411 E DEVICE EXCEPTION a1. SENSE=n2
```

It is important with this type of message that you tell the IBM Support Center the text of the message in full.

Note: Messages produced by the GDDM/TSO Print Utility and GDDM applications running in MVS/Batch or VSE/Batch appear on the system console log.

GDDM abend

This is APAR type 'ABENDUxxxx'.

A GDDM abend is issued when GDDM determines that it cannot continue. Some abend codes are produced from more than one module, and further problem analysis may be needed; the IBM Support Center staff can advise you.

A list of GDDM abends and the modules that issue them are listed in Appendix B, "Abend codes" on page 173.

If you are using GDDM-IVU, an abend subcode is contained in register 0. You should report this with the abend code.

System abend

This is APAR type 'ABENDSxxxx'.

A system abend is issued when the operating system determines that execution cannot continue. Report the module (CSECT) and the offset into that module at which the abend occurred.

If the subsystem produces a diagnostic message, that message normally incorporates the abend code. An exception is TSO where, *after* receiving the message: PROGRAM TERMINATED DUE TO ERROR+, you need to enter a question mark (?). TSO then produces a second-level message incorporating the abend code, if applicable. If you then press ENTER, TSO produces a dump. Note that a dump can only be produced if a SYSABEND or SYSUDUMP file has been previously allocated. You may have to run the application again and reproduce the problem.

Loop

This is APAR type 'LOOP'.

Standard host system and subsystem techniques or the GDDM trace function can be used to diagnose loop and wait states. If in doubt, use the trace facilities described in Chapter 2, "GDDM tracing" on page 21 to determine whether there is a loop.

Reduced performance

This is APAR type 'PERFM'.

Apparent poor performance by GDDM may be caused by the host computer system or subsystem. Report this as a GDDM problem only if you determine that GDDM is responsible, or if the performance problem occurs as the result of the installation of a PTF or of a new release.

Wait

This is APAR type 'WAIT'.

GDDM contains a small number of calls to system or subsystem wait services. It also issues requests for system services. These may produce wait states. Report this as a GDDM problem only if you determine that GDDM causes the wait.

GDDM documentation error

This is APAR type 'DOC'.

This covers incorrect, missing, or unclear information in the GDDM library.

Note: For missing messages, report a message problem. For errors in the utility panels and the GDDM-IMD tutorial, report an unexpected output problem.

If you have a documentation problem, report the complete title and order number for the book in error. For example, the title of the book you are reading is *GDDM Diagnosis*, and its order number is SC33-0870-00.

The GDDM library is shown on page 229.

Add the numbers of any Technical Newsletters (TNL) that have been applied to the books.

Defining the operating environment

Report the following four facts about the operating environment at the time that the problem occurred.

1. Program identifier.

GDDM comprises these licensed programs:

Program	Contents
GDDM/MVS, GDDM/VSE, GDDM/VM	<ul style="list-style-type: none">• Graphics and alphanumerics API• Image Symbol Editor• Run-time mapping• GDDM-REXX – use of GDDM from EXECs written for the VM/System Product Interpreter, and for the TSO subsystem.• GDDM-PCLK – use of GDDM on a personal computer system linked to a host computer• GDDM-OS/2 Link – use of GDDM under OS/2 on a personal computer system linked to a host computer.
GDDM-PGF	<ul style="list-style-type: none">• Presentation Graphics routines• Interactive Chart Utility• Vector Symbol Editor.
GDDM-IMD	Interactive creation of screen and printer maps at program development time.
GDDM-IVU	Viewing, creating, modifying, storing, and printing images.
GDDM-GKS	Use of GDDM with Graphical Kernel System functions.

In addition, national language support is provided enabling you to use one or more national languages for the GDDM Base programs, and for GDDM-PGF, GDDM-IVU, and GDDM-GKS. National language support for all these GDDM licensed programs is shipped on a single tape for each operating system environment. That is, there is a GDDM/MVS NLS tape, a GDDM/VM NLS tape, and a GDDM/VSE NLS tape.

When national language support is installed, it establishes the national language for any or all of the following:

- Messages for the GDDM Base programs

- Messages and panels for GDDM-PGF
- Messages and panels for GDDM-IVU
- Messages for GDDM-GKS

If you find a problem with the translation of a panel or a message, report the problem to your IBM representative, but do not submit an APAR.

2. Release level.

This is GDDM Version 3, Release 2. You can identify it as 'R32'.

3. Application.

Report the application being run. Identify it as one of the following types:

- A program written by the user
- A program written by a third party
- A program written by IBM, but not a GDDM utility
- A GDDM utility (ICU, Vector Symbol Editor, Image Symbol Editor, Print Utility, IVU, GDDM-PCLK, and GDDM-OS/2 Link)

4. Subsystem.

When running under MVS or MVS/XA, specify whether TSO, CICS, or IMS is being used.

Defining the devices in use

GDDM supports a wide range of devices. It is important that you report the device environment accurately. You should report:

- The device type and model
- The controller type and model to which the device is attached
- The device tokens, if used
- The device definitions
- The display type
- The display adapter type

An application always has a primary device opened. This may be a real device, or a dummy device. If it is a real device, report its device type and model and those of the controller to which it is attached. If it is a dummy device, report the device token. The application has a dummy device if it is running in a batch environment or on a disconnected VM machine.

Some applications may use device tokens even though they have a real primary device. Report this.

The application may also have an alternate device open. If there is an alternate device and it has a device token associated with it, report this token.

If a device token supplied by GDDM is being used, report the token name. If a user-defined token is being used, you may be asked to give the definitions for it.

If the symptom is associated with a GDDM family-1 printer, report the LU type of the printer.

On some subsystems, the device definitions to the subsystem or telecommunication access method are important:

- If TSO is being used, report the PSERVIC.
- If the GDDM/TSO Print Utility is being used, report the PSERVIC.
- If IMS is being used, report the device token being used by the system definition database.
- If CICS is being used, report the feature operands in the DFHTCT macro.
- If GDDM-PCLK is being used, report the following:
 - the emulator in use
 - the emulator level
 - the definitions on the GDDM-PCLK setup panels
- If GDDM-OS/2 Link is being used, report the following:
 - the OS/2 version – run SYSLEVEL from an OS/2 session
 - the GQFLEVEL information – run GQFLEVEL from an OS/2 session
 - the CSD level information – run SYSLEVEL from an OS/2 session
 - a copy of the CONFIG.SYS file
 - what national language you are using

Reporting the problem

When you have completed the problem report form, report the problem to the IBM Support Center staff. The action they take depends on whether a fix exists for the problem you report.

If there is no fix, they will give you guidance on what to do next.

Submitting an APAR

The IBM Support Center staff may ask you to submit an APAR on your GDDM problem. If they do, they will give you an *APAR number* and *severity code*. If necessary, they can help you to complete the form.

Appendix A. Data areas

Diagnosis, Modification or Tuning Information

This section lists the major GDDM control blocks that are likely to be produced in trace output. They are presented in alphabetic order of mnemonic. The description of each has three parts:

1. A reference list that precedes the detailed portion of the control block. The format of the reference list is:

Control block mnemonic and title

For example, ADMTAAB – application anchor block (AAB)

Function

A brief description of what the control block does.

2. An alphabetic index for all items contained in the control block.
3. A tabular description of the storage layout of the control block. The following information is provided:

Offset Field addresses in hexadecimal notation, relative to the beginning of the control block.

Type The type of item, one of:

A Address
 B Bit string
 C Character string
 F Fixed binary signed
 U Unsigned binary.

Bit positions and values in bit strings are shown in the following examples:

1... (a reference to bit 0)
11 (a reference to bits 6 and 7).

Length The length of the field in bytes, as a decimal value.

Name The name of the field.

Description A brief description of the meaning of the field.

GDDM object file formats are described in the *GDDM Base Application Programming Reference* book.

ADMTAAB – application anchor block (AAB)

Function

The application anchor block (AAB), contains information which is passed between the application and GDDM when the application is using the reentrant application call interface.

Specifically, it contains an anchor used by GDDM for all its cross-invocation storage; feedback areas for returning error codes back to the application; and an optional application-dependent extension.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
AABAP	4	AABFC	0	AABXNEPA	8
AABEC	2	AABSC	0	AABXTN	8
AABEXTD	0	AABXFEPA	C	ADMTAAB	0

Storage layout

Offset	Type	Length	Name	Description
0	C	8	ADMTAAB	Application anchor block
0	F	4	AABFC	GDDM fullword feedback code
0	F	2	AABSC	GDDM severity code
2	F	2	AABEC	GDDM error code
4	A	4	AABAP	GDDM anchor pointer
8	C	0	*	Application dependent extension
0	C	16	AABEXTD	Extended AAB used only for the GDDM allocated, static AAB used for the NACI
0	C	8	*	Normal AAB
8	C	8	AABXTN	AAB extension
8	A	4	AABXNEPA	Normal entry point address
C	A	4	AABXFEPA	Fast entry point address

ADMTACA – application interface control area

Function

This is the main anchor block of the application interface component (AIC). It contains the general AIC status, and pointers to other AIC data. There is one instance per GDDM representation. It is organized in two sections. The first section contains those fields that are initialized by GDDM stub linkage code. The second section contains those fields that are initialized by the subsystem-independent AIC controller code.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ACAAABP	3C	ACAIFCBP	54	ACASTXTP	D8
ACAACPX	A0	ACAIFTYP	4A	ACATRACC	86
ACAAPLTP	64	ACAINIT	3A	ACATRCEF	38
ACAATRNP	9C	ACAINSMN	D0	ACATRCEP	6C

Name	Offset	Name	Offset	Name	Offset
ACABLANG	87	ACAMEP0P	94	ACATRCFL	70
ACACFDTP	60	ACAMSTGI	D2	ACATRCCLC	7E
ACADSAOP	58	ACAMSTGL	C8	ACATRCCLP	88
ACADSAP	8	ACAMSTGP	E4	ACATRCNN	78
ACADSASP	4	ACAMSX0P	C	ACATRCPC	7C
ACADTRNP	98	ACANLCPX	AC	ACATRCQM	F0
ACAESSF	B0	ACA0BCPL	A4	ACATRCSE	84
ACAEXITH	90	ACA0BCPS	A8	ACATRCSP	80
ACAEXITP	8C	ACAPINSP	DC	ACATRCSTB	68
ACAEXPBP	C0	ACAPLPTR	44	ACATRC13	74
ACAEXTRP	E8	ACAPRDCT	CC	ACATR58L	F4
ACAEXT31	8C	ACAPRMNO	48	ACATXTMN	CE
ACAFLAGS	38	ACAPTXTTP	D4	ACAUCXAF	38
ACAFLG1	38	ACAPVTAP	34	ACAUETPS	BC
ACAFORCE	86	ACAPVTEP	10	ACAULANG	93
ACAGKEH	92	ACAQERB	100	ACAUR1	4C
ACAGKSEP	14	ACAQERBP	C4	ACAUR14	50
ACAGKSSP	EC	ACARCPP	40	ADMTACA	0
ACAICBP	B4	ACASINSP	E0		
ACAID	0	ACASOSSP	5C		

Storage layout

Offset	Type	Length	Name	Description
0	C	*	ADMTACA	AIC control area
0	C	4	ACAID	Eye catcher 'ACA '
STUB-DEPENDENT SECTION				
4	A	4	ACADSASP	A(Initial DSA stack)
8	A	4	ACADSAP	A(Initial DSA)
C	A	4	ACAMSX0P	A(Default error exit)
10	A	4	ACAPVTEP	A(ESI product vector table)
14	A	4	ACAGKSEP	A(GKS error handler)
18	A	4	* (7)	Spare fields
STUB-INDEPENDENT SECTION				
34	A	4	ACAPVTAP	A(AIC product vector table)
GENERAL FLAGS AND INDICATORS				
38	C	2	ACAFLAGS	General flags
38	C	1	ACAFLG1	Flag byte 1
	1... ..		*	Spare
	.1.. ..		ACATRCSE	Trace processing is open
	..1.		ACAUCXAF	User call exit is active
39	C	1	*	Reserved
3A	F	2	ACAINIT	Initialization indicator

Offset	Type	Length	Name	Description
APPLICATION CALL RELATED FIELDS				
3C	A	4	ACAAABP	A(AAB)
40	A	4	ACARCPP	A(Request control parameter)
44	A	4	ACAPLPTR	A(User parms, less AAB and RCP)
48	F	2	ACAPRMNO	No. of user parms
4A	F	2	ACAIFTYP	Invocation type 0 NACI, 1 RACI, 2 SPI
4C	F	4	ACAUR1	Invoker's register 1
50	F	4	ACAUR14	Invoker's register 14
HOUSEKEEPING RELATED FIELDS				
54	A	4	ACAIFCBP	A(Interface control block)
58	A	4	ACADSAOP	A(Overflow DSA stack)
5C	A	4	ACASOSSP	A(SOS stack)
SPARE FIELDS				
60	A	4	ACACFDTP	A(Call format descr table)
64	A	4	ACAAPLTP	A(APL equivalence table)
TRACE-RELATED FIELDS				
68	A	4	ACATRCTB	A(In core trace table)
6C	A	4	ACATRCEP	A(Trace processor)
70	A	4	ACATRCFL	A(Trace file request block)
74	A	4	ACATRC13	A('Current DSA')
78	F	4	ACATRCNN	Current trace sequence no.
7C	F	2	ACATRCPC	Current page count
7E	F	2	ACATRCLC	Current line count
80	A	4	ACATRCSP	A(Trace stg use tables)
84	F	2	ACATRCSE	Index of cur stg use entry
	1... ..		ACAFORCE	Force specified = 1
	.1... ..		ACATRACC	Trace string accepted = 1
	..11 1111		*	
87	C	1	ACABLANG	Backup language
88	A	4	ACATRCLP	A(Trace loop detect table)
				See extension fields for
				A(Ex trace control block)
ERROR PROCESSING RELATED FIELDS				
8C	A	4	ACAEXITP	A(User error exit)
	1... ..		ACAEXT31	AMODE of user exit
90	F	2	ACAEXITH	Current error exit threshold
	1... ..		ACAGKEH	GKS error handling in progress
	.111 1111		*	Spare fields
93	C	1	ACAULANG	Currently used language
94	A	4	ACAMEP0P	A(Error processor)
COUNTRY EXTENDED CODE PAGE (CECP) RELATED FIELDS				
98	A	4	ACADTRNP	A(GDDM ADMDATRN module)
9C	A	4	ACAATRNP	A(Current DATRN module)
A0	F	4	ACAACPX	Application code page index
A4	F	4	CAOBCPL	Object code page load
A8	F	4	CAOBCPS	Object code page save
AC	F	4	ACANLCPX	Natlang code page index

Offset	Type	Length	Name	Description
B0	C	1		
	1...		ACAESSF	ESSCPG API call flag
	.1111 1111		*	Spare
B1	C	3	*	Spare
B4	A	4	ACAICBP	A(ADMTICB)
B8	A	4	* (1)	Spare

FURTHER ERROR PROCESSING RELATED FIELDS

BC	A	4	ACAUETPS	A(User error text tables)
C0	A	4	ACAEXPBP	A(FSEXIT parameter block)
C4	A	4	ACAQERBP	A(Query error block)

FURTHER EXTENSION FIELDS

C8	F	4	ACAMSTGL	L(Merge storage block)
CC	F	2	ACAPRDCT	GDDM family product count
CE	F	2	ACATXTMN	Message text module count
D0	F	2	ACAINSMN	Message insert module count
D2	F	2	ACAMSTGI	Merge storage block index
D4	A	4	ACAPTXTTP	A(Prim msg text module tab)
D8	A	4	ACASTXTTP	A(Sec msg text module tab)
DC	A	4	ACAPINSP	A(Prim msg insert module tab)
E0	A	4	ACASINSP	A(Sec msg insert module tab)
E4	A	4	ACAMSTGP	A(Merge storage block)
E8	A	4	ACAEXTRP	A(Ex trace control block)
EC	A	4	ACAGKSSP	A(GKS DSA stack)

EXTRA FIELDS FOR ENHANCED TRACE

F0	A	4	ACATRCQM	A(Quad max. Last quad)
F4	F	4	ACATR58L	Length for 5080 parm trace

POINTERS FOR FURTHER FUNCTIONS NOT YET IMAGINED

F8	A	4	* (2)	Spare fields
----	---	---	-------	--------------

FOR EASE OF CONTROL BLOCK EXTENSION, THE FOLLOWING FIELD IS ALWAYS LAST, AND NORMALLY ADDRESSED VIA THE FIELD ACAQERBP

100	C	*	ACAQERB	Query error block (see ADMTQERB)
-----	---	---	---------	----------------------------------

ADMTQCQYV – VM display communication block

Function

This control block contains the control information that is used by the VM input/output routine.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTQCQYV	0	CQYDVSTT	22	CQYPSCSW	50
CQYDADSP	2A	CQYDVTYP	21	CQYPSDTA	60
CQYDARMT	2A	CQYD8CL	34	CQYPSLCC	50
CQYDATTN	2B	CQYD8CP	38	CQYPSNSB	60
CQYDATTR	2A	CQYHDLEN	14	CQYPSNSC	60
CQYDAVCN	2A	CQYHEAD	10	CQYPSNSD	60
CQYDDISC	2B	CQYHPLEN	10	CQYPSNSE	60
CQYDEV	18	CQYPATH	40	CQYPSNSI	60
CQYDLLEN	28	CQYPATTN	58	CQYPSNSN	60
CQYDNUMB	1C	CQYPBUSY	58	CQYPSNSO	60
CQYDQR	2C	CQYPCC	50	CQYPSNSU	60
CQYDQRCL	2E	CQYPCCCK	59	CQYPSTMD	58
CQYDQREC	2C	CQYPCCW	54	CQYPSUB	51
CQYDQREH	2C	CQYPCDCK	59	CQYPUNCK	58
CQYDQRFL	2C	CQYPCHCK	59	CQYPUNEX	58
CQYDQRPN	2D	CQYPCHEN	58	CQYPUST	58
CQYDQRPS	2C	CQYPCST	59	CQYPXWRD	44
CQYDQRRW	30	CQYPCUE	58	CQYSECT	10
CQYDQR14	2C	CQYPDVEN	58	DISPASYF	2
CQYDQYCD	32	CQYPEXIT	40	DISPBUFF	9
CQYDRCLS	24	CQYPFLG	48	DISPCLR	2
CQYDREAL	24	CQYPICCK	59	DISPCMD	8
CQYDRFTR	27	CQYPICL	59	DISPDEVA	0
CQYDRMDL	26	CQYPKSL	50	DISPFLAG	2
CQYDRTYP	25	CQYPLIO	48	DISPNORD	4
CQYDSTAT	2B	CQYPLOG	50	DISPOPCC	C
CQYDTMCD	29	CQYPORB	51	DISPOPER	8
CQYDUSCT	18	CQYPPCI	59	DISPPA1	2
CQYDVCLS	20	CQYPPGCK	59	DISPRCDE	3
CQYDVCNS	28	CQYPPRCK	59	DISPRW	8
CQYDVFLG	23	CQYPRCT	5A	DISPSIZE	E
CQYDVIRT	20	CQYPSCNT	5C	DISPWAIT	2

Storage layout

Offset	Type	Length	Name	Description
0			ADMTQCQYV	Display communications block
0	F	2	DISPDEVA	Device address (-1 = console)
2	B	1	DISPFLAG	Option byte
	1...		DISPCLR	Clear screen before write
	.1..		DISPWAIT	Wait after write
	..1.		DISPPA1	Caller wants to see PA1
	...1		DISPASYF	Asynch I/O received
 1111		*	
3	F	1	DISPRCDE	Return code for snap tracing
4	F	2	DISPNORD	No. of bytes read/written
6	F	2	*	Not used

Offset	Type	Length	Name	Description
8	C	8	DISPOPER	Start of operation list
8	B	1	DISPCMD	3270 local command code
	1111 111.		*	
1		DISPRW	1 if write command
9	A	3	DISPBUFF	Address of read/write buffer
C	F	2	DISPOPCC	CCW flags and CTL
E	F	2	DISPSIZE	Length of read/write buffer

Mapping of CQYSECT for Console Query function

10			CQYSECT	Query reply buffer
10			CQYHEAD	Reply buffer header
10	F	4	CQYHPLEN	Length of path section
14	F	4	CQYHDLEN	Length of device section
18			CQYDEV	Device section
18	F	4	CQYDUSCT	No. of paths opened to this device
1C	F	4	CQYDNUMB	Virtual device number
20			CQYDVIRT	Virtual device information
20	C	1	CQYDVCLS	Virtual device type class
21	C	1	CQYDVTYP	Virtual device type
22	C	1	CQYDVSTT	Virtual device status
23	C	1	CQYDVFLG	Virtual device flags
24			CQYDREAL	Real device info
24	C	1	CQYDRCLS	Real device type class
25	C	1	CQYDRTYP	Real device type
26	C	1	CQYDRMDL	Real model number
27	C	1	CQYDRFTR	Real feature code
28			CQYDVCNS	More diag x'24' info
28	C	1	CQYDLLEN	Line length
29	C	1	CQYDTMCD	Terminal code
2A	B	1	CQYDATTR	Device attribute flag
	1111 1...		*	Reserved
1..		CQYDARMT	Device is a remote 3270
1.		CQYDADSP	Device is a display
1		CQYDAVCN	Device is a virtual console
2B	B	1	CQYDSTAT	Device status flag 1
	1...		CQYDATTN	Attention pending
	.1..		CQYDDISC	Device is disconnected
	..11 1111		*	Reserved
2C			CQYDQQR	First 6 bytes 8C info
2C	B		CQYDQQRFL	Flags
	1...		CQYDQREC	Extended color
	.1..		CQYDQREH	Extended highlighting
	..1.		CQYDQRPS	PSS
	...1 111.		*	Not used
1		CQYDQR14	14-bit addressing
2D	C	1	CQYDQRPN	Number of partitions
2E	F	2	CQYDQRCL	Number of columns
30	F	2	CQYDQRRW	Number of rows
32	C	1	CQYDQYCD	Device query code
33	C	1	*	Reserved
34	F	4	CQYD8CL	Length of remaining 8C info
38	A	4	CQYD8CP	Ptr to WSF info after first 6 bytes of 8C info
3C	F	4	*	Reserved
40			CQYPATH	Path data section
40	A	4	CQYPEXIT	User exit address
44	F	4	CQYPXWRD	User word
48	B	1	CQYPFLG	Path flag

Offset	Type	Length	Name	Description
	1... ..		CQYPLIO	Path did last I/O
	.111 1111		*	Reserved
49	C	3	*	Reserved
4C	F	4	*	Reserved
50			CQYPSCSW	Channel status word
50	B	1	CQYPSLCC	Logout pend/cond codes
	1111 1...		CQYPKSL	Key/suspend/log bits
1..		CQYPLOG	Logout pending
11		CQYPCC	Deferred condition code
51	1111 1...		CQYPORB	Flags from orb
111	19 bits	CQYPSUB	Subchannel control bits
	1111 1111			
	1111 1111			
54	A	4	CQYPCCW	Last CCW executed
58	B	1	CQYPUST	Unit status byte
	1... ..		CQYPATTN	Attention
	.1.. ..		CQYPSTMD	Status modifier
	..1.		CQYPCUE	Control unit end
	...1		CQYPBUSY	Busy
	... 1...		CQYPCHEN	Channel end
1..		CQYPDVEN	Device end
1.		CQYPUNCK	Unit check
1		CQYPUNEX	Unit exception
59	B	1	CQYPCST	Channel status byte
	1... ..		CQYPPCI	Program-controlled int.
	.1.. ..		CQYPICL	Incorrect length
	..1.		CQYPPGCK	Program check
	...1		CQYPPRCK	Protection check
	... 1...		CQYPCDCK	Channel data check
1..		CQYPCCK	Channel control check
1.		CQYPICCK	Interface control check
1		CQYPCHCK	Chaining check
5A	F	2	CQYPRCT	Residual count
5C	F	4	CQYPSCNT	Sense count
60	C	32	CQYPSDTA	Sense data
	1... ..		CQYPSNSC	Command reject
	.1.. ..		CQYPSNSI	Intervention required
	..1.		CQYPSNSB	Bus-out check
	...1		CQYPSNSE	Equipment check
	... 1...		CQYPSNSD	Data check
1..		CQYPSNSU	Unit specify
1.		CQYPSNSN	Control check
1		CQYPSNSO	Operation check
61	C	31	*	Remainder of sense data

ADMTDFT – general defaults table

Function

This control block contains the current settings of all GDDM defaults and user exits. It also contains a pointer to the internal list of nickname entries; see “ADMTNICK – nickname list entry format” on page 164.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
------	--------	------	--------	------	--------

ADMTDFT - General defaults table

ADMTDFT	0	DFT0ERL	48	DFT0NSNF	32
DFTH	0	DFT0ERP	44	DFT0OBJO	24
DFTHHEAD	0	DFT0ERTH	3C	DFT0OPNF	32
DFTHLENG	4	DFT0ERTY	40	DFT0PA2F	32
DFTHUDS	8	DFT0ESIF	30	DFT0PVCF	31
DFTHUDSL	8	DFT0FFEF	31	DFT0REMF	32
DFTHUDSP	C	DFT0FFSF	31	DFT0SNAF	32
DFTXTNA	E0	DFT0FRCV	D8	DFT0SOSF	1C
DFT0	30	DFT0FSXK	1C	DFT0SVBZ	5C
DFT0ABRF	31	DFT0FSXP	18	DFT0SYNF	31
DFT0APCP	EC	DFT0GKWT	90	DFT0SYSO	2C
DFT0AUKB	31	DFT0GSXK	14	DFT0TFMT	36
DFT0CIBL	64	DFT0GSXP	10	DFT0TRBZ	58
DFT0CIBP	68	DFT0ICUF	DA	DFT0TRCE	4C
DFT0CLTB	108	DFT0ICUI	3A	DFT0TRCN	50
DFT0CMFP	31	DFT0ICUP	D9	DFT0TRLI	D4
DFT0CPG4	88	DFT0ICUS	DB	DFT0TRNP	DC
DFT0CTL5	D4	DFT0INCP	E8	DFT0TRWD	D0
DFT0DBDG	D4	DFT0LANG	34	DFT0TSHR	D4
DFT0DBDV	100	DFT0LCLF	32	DFT0TYPF	110
DFT0DBDY	D4	DFT0MGST	60	DFT0TYP5	10C
DFT0DBLN	39	DFT0MSOC	38	DFT0UXBO	28
DFT0DBMX	78	DFT0MSOF	31	DFT0XFBF	32
DFT0DFMT	35	DFT0NFMT	37	DFT0XFBF	70
DFT0DFSF	30	DFT0NICK	6C	DFT0XFBP	74
DFT0EINP	F0	DFT0NLSF	30	DFT00	10

ADMTDFTO - Object type defaults (Offset defined by DFT0OBJO)

ADMTDFTO	0	DFT0OBJT	0
----------	---	----------	---

ADMTDFTU - User exit blocks (Offset defined by DFT0UXBO)

ADMTDFTU	0	DFTUXCIK	10	DFTUXTSC	0
DFTUXCI	C	DFTUXCIP	14	DFTUXTSK	4
DFTUXCIC	C	DFTUXTS	0	DFTUXTSP	8

ADMTDFTC - CICS-specific defaults (Offset defined by DFT0SYSO)

ADMTDFTC	0	DFTCIFMT	30	DFTCSYSP	14
DFTCDECK	8	DFTCPRAP	20	DFTCTISF	0
DFTCDFPX	24	DFTCPRNT	10	DFTCTQRY	40
DFTCIADS	18	DFTCSRAP	1C	DFTCTRTD	4
DFTCIESL	28	DFTCSTGF	38	DFTCTSPX	C

Name	Offset	Name	Offset	Name	Offset
------	--------	------	--------	------	--------

ADMTDFTI - IMS-specific defaults (Offset defined by DFT0SYSO)

ADMTDFTI	0	DFTIOBDK	98	DFTISHUT	60
DFTICHRT	48	DFTIOBDS	88	DFTISYSP	70
DFTIDBXX	80	DFTIOBRK	90	DFTITRDD	20
DFTIDECK	28	DFTIOBRS	80	DFTIUIOL	10
DFTIEXIT	58	DFTIPRNT	30	DFTIUMAX	C
DFTIISSE	38	DFTISDSD	50	DFTIVSSE	40
DFTIMAST	68	DFTISDSF	A8	DFTIWTOD	18
DFTIMODN	78	DFTISDSS	A0	DFTIWTOR	14

ADMTDFTT - TSO-specific defaults (Offset defined by DFT0SYSO)

ADMTDFTT	0	DFTTEMUL	0	DFTTPRNT	28
DFTTADV	0	DFTTIADS	38	DFTTSYSP	30
DFTTCOLM	50	DFTTICBP	4	DFTTS99S	8
DFTTCPT	68	DFTTIESL	20	DFTTS99U	60
DFTTDECK	18	DFTTIFMT	40	DFTTTRDD	10
DFTTDFDD	58	DFTTMONO	48		

ADMTDFTV - CMS-specific defaults (Offset defined by DFT0SYSO)

ADMTDFTV	0	DFTVDFFT	68	DFTVPRNF	18
DFTVADV	0	DFTVIADS	28	DFTVSYSP	20
DFTVCOLM	48	DFTVIFMT	30	DFTVTMFT	8
DFTVCPT	70	DFTVMONO	40	DFTVTRFN	50
DFTVDECK	10	DFTVMSLT	38	DFTVTRFT	58
DFTVDFFN	60				

ADMTDFTD - VSE-specific defaults (Offset defined by DFT0SYSO)

ADMTDFTD	0	DFTDDFDD	18	DFTDTRDD	0
DFTDCOLM	10	DFTDMONO	8		

Storage layout

Offset	Type	Length	Name	Description
--------	------	--------	------	-------------

ADMTDFT - General defaults table

0	C	248	ADMTDFT	General defaults table
0	C	16	DFTH	Header
0	C	4	DFTHHEAD	'DFT 'eye catcher
4	F	4	DFTHLENG	Length of defaults table
8	C	8	DFTHUDS	Dummy UDS information
8	F	4	DFTHUDSL	Dummy length (non zero)
C	A	4	DFTHUDSP	Dummy address(zero)
10	C	32	DFT00	Storage exit defaults
10	A	4	DFT0GSXP	Addr(Get stg appl exit)
14	F	4	DFT0GSXK	Appl defined get storage token
18	A	4	DFT0FSXP	Addr(Free stg appl exit)
1C	F	4	DFT0FSXK	Appl defined free storage token
20	1...		DFT0SOSF	Short on storage processing
	.1..		*	
	..1.		*	
	...1		*	
 1...		*	

Offset	Type	Length	Name	Description
1..		*	
1..		*	
1..		*	
21	C	1	*	
22	C	1	*	
23	C	1	*	
24	F	4	DFT0OBJO	Offset to object type defaults
28	F	4	DFT0UXBO	Offset to user exit blocks
2C	F	4	DFT0SYSO	Offset to subsystem specific dtf
30	C	200	DFT0	Subsystem common defaults
30	1... ..		DFT0ESIF	Initialize ESI only
	.1.. ..		DFT0DFSF	DFT file DFTS on SPIB
	..1.		DFT0NLSF	Natlang DFTS on SPIB
	...1		*	
 1...		*	
1..		*	
1..		*	
1..		*	
31	1... ..		DFT0PVCF	SPI parameter verification
	.1.. ..		DFT0ABRF	Return on abend
	..1.		DFT0SYNF	Synchronized I/O
	...1		DFT0CMPF	PS compression allowed
 1...		DFT0AUKB	Always unlock keyboard
1..		DFT0FFSF	Form feed at start
1..		DFT0FFEF	Form feed at end
1..		DFT0MSOF	Use SO/SI in mixed fields
32	1... ..		DFT0LCLF	All devices are local
	.1.. ..		DFT0REMF	All devices are remote
	..1.		DFT0SNAF	All devices are SNA
	...1		DFT0NSNF	All devices are non-SNA
 1...		*	
1..		DFT0OPNF	Must open device on FSINIT
1..		DFT0PA2F	Return PA2 to ASREAD(VM)
1..		DFT0XFBF	Attn FBACK block set(VM)
33	C	1	*	
34	C	1	DFT0LANG	National language character
35	F	1	DFT0DFMT	Date convention
36	F	1	DFT0TFMT	Time convention
37	F	1	DFT0NFMT	Number convention
38	C	1	DFT0MSOC	SO/SI emulation character
39	C	1	DFT0DBLN	DBCS symbol set language char
3A	F	1	DFT0ICUI	ICU isolate value
3B	C	1	*	
3C	F	4	DFT0ERTH	Error threshold value
40	F	4	DFT0ERTY	Type of error exit action
44	A	4	DFT0ERP	A(F/BACK block)
48	F	4	DFT0ERL	L(F/BACK block)
4C	F	4	DFT0TRCE	Trace control value
50	F	4	DFT0TRCN	Incore trace table size
54	C	4	*	
58	F	4	DFT0TRBZ	Transmission buffer size
5C	F	4	DFT0SVBZ	FSSAVE buffer size
60	F	4	DFT0MGST	Mapgroup storage threshold
64	F	4	DFT0CIBL	L(Call info feedback block)
68	A	4	DFT0CIBP	A(Call info feedback block)
6C	A	4	DFT0NICK	Nickname list chain anchor
70	F	4	DFT0XFBL	Attention FBACK block length
74	A	4	DFT0XFBP	Attention FBACK block address

Offset	Type	Length	Name	Description
78	F	4	DFT0DBMX	DBCS component in core threshold
7C	C	4	*	
80	C	8	*	
88	C	8	DFT0CPG4	Default code page name for 4250
90	C	8	DFT0GKWT	(8) GKS wkstation DFLT DVC tokens
D0	F	4	DFT0TRWD	Trace output width
D4	A	4	DFT0TRLI	Trace list chain anchor
D8	1... ..		DFT0TSHR	Trace file sharing
	.1.. ..		DFT0DBDG	GDDM determined DBCS support
	..1.		DFT0DBDY	DBCS supported
	...1		DFT0CTLS	Default CLTSAVE value
 1...		DFT0FRCV	Force HPA validation
1..		*	
1.		*	
1		*	
D9	F	1	DFT0ICUP	ICU panel color
DA	F	1	DFT0ICUF	ICU format defaults
DB	F	1	DFT0ICUS	ICU symbol set use
DC	A	4	DFT0TRNP	Translate table address
E0	A	4	DFTXTNA	Address of defaults extension block
E4	C	4	*	
E8	F	4	DFT0INCP	Installation code page
EC	F	4	DFT0APCP	Application code page
F0	1... ..		DFT0EINP	Enable CECF input
	.111 1111		*	
F1	C	7	*	
F8	C	8	DFT0DBDI	DBCS default image symbol set
100	C	8	DFT0DBDV	DBCS default vector symbol set
108	A	4	DFT0CLTB	Addr (color mapping table)
10C	A	4	DFT0TYP5	Addr (symbol set to font mapping)
110	A	4	DFT0TYPF	Addr (presentation font to font mapping)
104	C	4	*	

ADMTDFTO - Object type defaults (Offset defined by DFT0OBJO)

0	C	96	DMTDFTO	Object type defaults
0	C	8	DFT0OBJT	(12) Object type table

ADMTDFTU - User exit blocks (Offset defined by DFT0UXBO)

0	C	24	DMTDFTU	User exit blocks
0	C	12	DFTUXTS	Task switch exit UXBLOCK
0	F	4	DFTUXTSC	User exit code
4	F	4	DFTUXTSK	User exit token
8	A	4	DFTUXTSP	User exit address
C	C	12	DFTUXCI	Call intercept exit UXBLOCK
C	F	4	DFTUXCIC	User exit code
10	F	4	DFTUXCIK	User exit token
14	A	4	DFTUXCIP	User exit address

ADMTDFTC - CICS-specific defaults (Offset defined by DFT0SYSO)

0	C	68	DMTDFTC	CICS specific defaults
0	1... ..		DFTCTISF	Use trans independent services
	.111 1111		*	
1	C	1	*	
2	C	1	*	
3	C	1	*	

Offset	Type	Length	Name	Description
4	C	4	DFTCTRTO	CICS trace td name
8	C	4	DFTCDECK	CICS deck o/p td name
C	C	4	DFTCTSPX	CICS temp storage prefix
10	C	4	DFTCPRNT	CICS GDDM print utility name
14	C	4	DFTCSYSP	CICS system printer TD name
18	C	4	DFTCIADS	CICS IMD ADS O/P TD name
1C	A	4	DFTCSRAP	CICS stg audit trail anchor
20	A	4	DFTCPRAP	CICS pgm audit trail anchor
24	C	4	DFTCDFPX	CICS defaults TS prefix
28	C	8	DFTCIESL	CICS IMD ADMGIMP FCT name
30	C	8	DFTCIFMT	CICS IMD staged data filetype
38	C	8	DFTCSTGF	CICS IMD staged file FCT name
40	C	4	DFTCTQRY	CICS temp storage prefix for QR

ADMTDFTI - IMS-specific defaults (Offset defined by DFT0SYSO)

0	C	176	ADMTDFTI	IMS-specific defaults
0	1111 1111		*	
1	C	3	*	
4	A	4	*	
8	A	4	*	
C	F	4	DFTIUMAX	IMS GDDM util max no of users
10	F	4	DFTIUIOL	IMS GDDM input area size
14	C	4	DFTIWTOR	IMS WTO routing codes
18	C	4	DFTIWTOD	IMS WTO descriptor codes
1C	C	4	*	
20	C	8	DFTITRDD	IMS trace DD name
28	C	8	DFTIDECK	IMS deck O/P LTERM name
30	C	8	DFTIPRNT	IMS GDDM print utility name
38	C	8	DFTIISSE	IMS ISE transaction name
40	C	8	DFTIVSSE	IMS VSE transaction name
48	C	8	DFTICHRD	IMS ICU transaction name
50	C	8	DFTISDSD	IMS GDDM sysdef DBD name
58	C	8	DFTIEXIT	IMS util exit string
60	C	8	DFTISHUT	IMS util shutdown string
68	C	8	DFTIMAST	IMS util shutdown ITERM name
70	C	8	DFTISYSP	IMS system printer DEST name
78	C	8	DFTIMODN	IMS GDDM MFS mod name
80	C	48	DFTIDBXX	IMS segment/key field names
80	C	8	DFTIOBRS	Object DB root segment name
88	C	8	DFTIOBDS	Object DB dep segment name
90	C	8	DFTIOBRK	Object DB root key field name
98	C	8	DFTIOBDK	Object DB dep segment key field name
A0	C	8	DFTISDSS	Sysdef DB segment name
A8	C	8	DFTISDSF	Sysdef DB key field name

ADMTDFTT - TSO-specific defaults (Offset defined by DFT0SYSO)

0	C	112	ADMTDFTT	TSO specific defaults
0	1... ..		DFTTADV	Non quer 327X 2 is 3278/9
	.1... ..		DFTTEMUL	TSO emulation
	..11 1111		*	
1	C	1	*	
2	C	1	*	
3	C	1	*	
4	A	4	DFTTICBP	TSO print utility ICB
8	F	4	DFTTS99S	TSO SVC99 allocation size
C	C	4	*	

Offset	Type	Length	Name	Description
10	C	8	DFTTTRDD	TSO trace DD name
18	C	8	DFTTDECK	TSO deck O/P DD name
20	C	8	DFTTIESL	TSO IMD ADMGIMP DD name
28	C	8	DFTTPRNT	TSO print data set qualifier
30	C	8	DFTTSYSP	TSO system printer DD name
38	C	8	DFTTIADS	TSO IMD ADS DD name
40	C	8	DFTTIFMT	TSO IMD export DD name
48	C	8	DFTTMONO	TSO HRIG mono DD or DSNAME HLQ
50	C	8	DFTTCOLM	TSO HRIG col DD or DSNAME HLQ
58	C	8	DFTTDFDD	TSO defaults file DDNAME
60	C	8	DFTTS99U	TSO SVC99 unit specification
68	C	8	DFTTCPT	TSO CGM profile data set qualifier

ADMTDFTV - CMS-specific defaults (Offset defined by DFT0SYSO)

0	C	120	ADMTDFTV	CMS specific defaults
	1...		DFTVADVF	Non quer 328X is 3287 type
	.111 1111		*	
1	C	1	*	
2	C	1	*	
3	C	1	*	
4	C	4	*	
8	C	8	DFTVTMFT	CMS work file file-type
10	C	8	DFTVDECK	CMS deck O/P file-type
18	C	8	DFTVPRNF	CMS print file-type
20	C	8	DFTVSYSP	CMS system printer file-type
28	C	8	DFTVIADS	CMS IMD ADS file-type
30	C	8	DFTVIFMT	CMS IMD export file-type
38	C	8	DFTVMSLT	CMS IMD MSL file-type
40	C	8	DFTVMONO	CMS HRIG mono file-type
48	C	8	DFTVCOLM	CMS HRIG col file-type
50	C	8	DFTVTRFN	CMS trace file-name
58	C	8	DFTVTRFT	CMS trace file-type
60	C	8	DFTVDFFN	CMS defaults file-name
68	C	8	DFTVDFFT	CMS defaults file-type
70	C	8	DFTVCPT	CMS CGM profile data set qualifier

ADMTDFTD - VSE-specific defaults (Offset defined by DFT0SYSO)

0	C	32	ADMTDFTD	VSE specific defaults
0	C	8	DFTDTRDD	VSE trace D label
8	C	8	DFTDMONO	VSE HRIG mono D label
10	C	8	DFTDCOLM	VSE HRIG color D label
18	C	8	DFTDDFDD	VSE defaults file D label

ADMTDSA – dynamic save area (DSA)

Function

This control block contains a GDDM dynamic save area (DSA).

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTDSA	0	DSALEN	2	DSASOSP	4C
DSAEPTR	50	DSANEXT	8	DSASTATE	48
DSAFLAGS	0	DSARSA	C	DSATRCE	48
DSAHDR	48	DSASOSEQ	4C	DSAWORD1	0
DSALAST	4				

In the DSA extension (addressed from DSAEPTR):

ADMTDSAE	0	DSAEEND	18	DSANEST	4
DSACNTRL	4	DSAERRPL	C	DSATRCVS	8
DSAEDEAE	0	DSAERRPP	10		

Storage layout

Offset	Type	Length	Name	Description
0		88	ADMTDSA	Standard DSA declaration
0	C	4	DSAWORD1	
0	C	2	DSAFLAGS	Reserved set to x'0000' first byte is committed to always remain zero
2	F	2	DSALEN	Allocated DSA length
4	A	4	DSALAST	A(Last DSA)
8	A	4	DSANEXT	A(Next DSA)
C	A	60	DSARSA	Register save area
48	C	8	DSASTATE	State controls propagated down via ADMMBEG macro
48	A	4	DSAHDR	A(DSA stack header)
	1...		DSATRCE	Trace optimization bit
	.111 1111		*	
4C	A	4	DSASOSP	A(Short on storage element)
	1...		DSASOSEQ	SOS event in progress
	.111 1111		*	
50	A	4	DSAEPTR	Pointer to DSA extension
54	A	4	*	Reserved
58	C	*	*	User area

In the DSA extension (addressed from DSAEPTR):

0	C	*	ADMTDSAE	Extension block
0	C	4	DSAEDEAE	Room for an eye catcher later
4	C	4	DSACNTRL	Standard DSA controls
4	F	1	DSANEST	DSA nesting level
5	C	3	*	Reserved
8	A	4	DSATRCVS	R14/R15 trace save slot
C	A	4	DSAERRPL	Error stack pointer for calls
10	A	4	DSAERRPP	Error stack pointer for process
14	C	4	*	Reserved
18	C	*	DSAEEND	End of extension stack

ADMTDSAO – dynamic save area overflow stack (DSAO)

Function

This control block contains the header of the GDDM dynamic save area overflow stack. This stack is used to supply “emergency” save areas to the DSA overflow processing code.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTDSAO	0	DSASOENT	8	DSASOMN	10
DSASOACA	C	DSASOHDR	0	DSASOW1	0
DSASOBOS	4				

Storage layout

Offset	Type	Length	Name	Description
0	C	16	ADMTDSAO	DSA overflow stack
0	C	16	DSASOHDR	DSA stack header
0	A	4	DSASOW1	Reserved
4	A	4	DSASOBOS	A(Bottom of stack)
8	A	4	DSASOENT	A(Out of line prolog)
C	A	4	DSASOACA	A(AIC control area)
10	C	*	DSASOMN	Allocation area

ADMTDSAS – dynamic save area stack (DSAS)

Function

This control block contains the header of a GDDM dynamic save area stack.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTDSAS	0	DSASENTE	8	DSASMAIN	10
DSASACAP	C	DSASHDR	0	DSASNXT	0
DSASBOS	4				

Storage layout

Offset	Type	Length	Name	Description
0	C	16	ADMTDSAS	DSA stack
0	C	16	DSASHDR	DSA stack header
0	A	4	DSASNXT	Chain to next DSA stack (0 if last)
4	A	4	DSASBOS	A(Bottom of stack)
8	A	4	DSASENTE	A(Out of line prolog)
C	A	4	DSASACAP	A(AIC control area)
10	C	*	DSASMAIN	Allocation area

ADMTIFCB – interface control block (IFCB)

Function

This is the interface control block (IFCB) which contains control information used throughout GDDM. Specifically, it contains the IFC response field, the interface control table (IFCT), and a subsystem-dependent subsystem save area.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTIFCB	0	IFCTGKSE	E0	IFCTPLT0	98
IFCBFC	0	IFCTGKSS	E4	IFCTPREV	5C
IFCBSSA	110	IFCTGKS0	E0	IFCTPRI	15
IFCTACOP	1C	IFCTGRA	15	IFCTPRVT	6C
IFCTAICE	70	IFCTHEAD	4	IFCTPTN	15
IFCTAICS	74	IFCTIMCE	E8	IFCTPTS	15
IFCTAIC0	70	IFCTIMCS	EC	IFCTPTXT	6C
IFCTA31C	14	IFCTIMC0	E8	IFCTPVSE	6D
IFCTA31F	14	IFCTIMDE	D0	IFCTROUT	15
IFCTBGSE	90	IFCTIMDS	D4	IFCTSCBP	8
IFCTBGSS	94	IFCTIMD0	D0	IFCTSOSE	20
IFCTBGS0	90	IFCTIMSE	C0	IFCTSPIP	24
IFCTCDUE	F0	IFCTIMSS	C4	IFCTSSMA	40
IFCTCDUS	F4	IFCTIMS0	C0	IFCTSSMT	3C
IFCTCDU0	F0	IFCTISEE	A0	IFCTSTOP	16
IFCTCOMP	44	IFCTISES	A4	IFCTTDF	10
IFCTCSIE	A8	IFCTISE0	A0	IFCTTDFA	10
IFCTCSIS	AC	IFCTIVUE	F8	IFCTTDFB	10
IFCTCSI0	A8	IFCTIVUS	FC	IFCTTDFD	10
IFCTDB22	16	IFCTIVU0	F8	IFCTTDFE	10
IFCTDDBG	48	IFCTMAXX	18	IFCTTDFI	10
IFCTDFDB	16	IFCTMODE	0	IFCTTDFK	11
IFCTDFTP	4C	IFCTMODS	4	IFCTTDFL	10
IFCTEDFP	50	IFCTMODX	0	IFCTTDFN	11
IFCTEFBP	28	IFCTNEXT	58	IFCTTDFO	11
IFCTERRA	2C	IFCTNL22	17	IFCTTDFP	10
IFCTERRL	34	IFCTNONN	14	IFCTTDFV	11
IFCTERRN	38	IFCTNSRE	C8	IFCTTDFY	11
IFCTERRP	30	IFCTNSRS	CC	IFCTTDF1	11
IFCTESIE	80	IFCTINSR0	C8	IFCTTDF3	11
IFCTESIS	84	IFCTOUTE	B8	IFCTTFRC	10
IFCTESI0	80	IFCTOUTS	BC	IFCTTLVL	13
IFCTFAPG	68	IFCTOUT0	B8	IFCTTQUL	12
IFCTFIGE	D8	IFCTPAG	15	IFCTTRCB	6C
IFCTFIGS	DC	IFCTPAPG	64	IFCTTRCE	10
IFCTFIG0	D8	IFCTPCLK	6D	IFCTTSID	78
IFCTFINT	14	IFCTPCGM	6D	IFCTTSIS	7C
IFCTFLGS	14	IFCTPCSF	6C	IFCTTSI0	78
IFCTFLG1	16	IFCTPDS	6C	IFCTUDFP	54
IFCTFPA	15	IFCTPDSO	6C	IFCTVERS	C
IFCTFSME	88	IFCTPGTP	60	IFCTVER1	C
IFCTFSMS	8C	IFCTPGDP	6C	IFCTVER2	E
IFCTFSM0	88	IFCTPHRG	6D	IFCTVSEE	B0
IFCTFTRM	14	IFCTPID	6C	IFCTVSES	B4
IFCTFWAP	14	IFCTPIM	6C	IFCTVSE0	B0
IFCTGEN	78	IFCTPLTE	98	IFCTWIN	15

Name	Offset	Name	Offset	Name	Offset
IFCTGKER	14	IFCTPLTS	9C	IFCT31LE	14
				IFCT5080	6C

Storage layout

Offset	Type	Length	Name	Description
0	C	*	ADMTIFCB	Interface control block.
0	F	2	IFCBFC	Feedback code
2	F	2	*	Reserved interface control table
4	C	4	IFCTHEAD	'IFCT' eye catcher
8	A	4	IFCTSCBP	A(Stub communications block)
C	C	4	IFCTVERS	Version fields
C	C	2	IFCTVER1	'Resident code' version
E	C	2	IFCTVER2	'Transient code' version
10	F	4	IFCTTRCE	Trace word
10	B	2	IFCTTDF	Trace flags
	1... ..		IFCTTFRC	1 = Suppress any further trace word changes component disablement flags
	.1.. ..		IFCTTDFA	1 = Disable AIC trace
	..1.		IFCTTDFL	1 = Disable TSI trace
	...1		IFCTTDFE	1 = Disable ESI trace
 1...		IFCTTDFD	1 = Disable FSM trace
1..		IFCTTDFB	1 = Disable BGS trace
1.		IFCTTDFP	1 = Disable PLT trace
1		IFCTTDFI	1 = Disable ISE trace
	1... ..		IFCTTDFY	1 = Disable CSI trace
	.1.. ..		IFCTTDFV	1 = Disable VSE trace
	..1.		IFCTTDFO	1 = Disable OUT trace
	...1		IFCTTDFK	1 = Disable IMS trace
 1...		IFCTTDFN	1 = Disable NSR trace
1..		IFCTTDF1	1 = Disable IMD trace
1.		*	Reserved
1		IFCTTDF3	1 = Disable IMC trace
12	F	1	IFCTTQUL	Trace qualifier
13	F	1	IFCTTLVL	Trace level
14	B	1	IFCTFLGS	General purpose flags
	1... ..		IFCTFWAP	1 => Within application
	.1.. ..		IFCTA31F	1 => GDDM initialized in 31 bit mode
	..1.		IFCTA31C	1 => GDDM called in 31 bit mode
	...1		IFCT31LE	1 => 31 bit load error has occurred (attempt use module above 16meg but GDDM initialized in 24bit mode)
 1...		IFCTNONN	1 => Bypass nickname processing
1..		IFCTFTRM	Implicit FSTERM required
1.		IFCTFINT	Implicit FSINIT performed
1		IFCTGKER	GKS error state
15	B	1	IFCTROUT	FSM routing flags:
	1... ..		IFCTPRI	Primary device exists
	.1.. ..		IFCTPTS	Partition set exists
	..1.		IFCTPTN	Partition exists
	...1		IFCTPAG	Page exists
 1...		IFCTWIN	Window exists
1..		*	Reserved
1.		IFCTGRA	Graphics may fastpath
1		IFCTFPA	Fastpath is active
16	B	1	IFCTFLG1	General purpose flags

Offset	Type	Length	Name	Description
	1...		IFCTDFDB	1 => GDDM has determined that the 'DFT' terminal can support line O/P DBCS through subsystem
	.1..		IFCTDB22	Base national language is DBCS
	..1.		IFCTSTOP	Stop recursive abends
	...1		IFCTSSCP	Symbol set code page not equal to appl. code page flag
 1...		IFCTPTAX	Auxiliary device print file
1..		IFCTTOFL	TOFILE output
1.		IFCTENAB	ADMBBGS OS/390 enable
1		IFCTENAP	ADMPMAIN OS/390 enable
17	C	1	IFCTNL22	Base national language
18	F	4	IFCTMAXX	CMS/XA store req. limit
1C	A	4	IFCTAC0P	A(AIC controller)
20	A	4	IFCTSOSE	A(AIC SOS processor)
24	A	4	IFCTSPIP	A(SPIB copy)
28	A	4	IFCTEFBP	A(Error feedback block)
2C	A	4	IFCTERRA	A(Error list anchor)
30	A	4	IFCTERRP	A(End of error list)
34	A	4	IFCTERRL	A(Code of last error added)
38	F	4	IFCTERRN	Number of errors in list
3C	F	4	IFCTSSMT	Small storage size threshold
40	A	4	IFCTSSMA	Small storage pool table
44	A	4	IFCTCOMP	A(X instance comm table)
48	A	4	IFCTDDBG	Development debug anchor
4C	A	4	IFCTDFTP	A(General defaults table)
50	A	4	IFCTEDFP	A(Default definition module)
54	A	4	IFCTUDFP	A(User defaults module)
58	A	4	IFCTNEXT	Next IFCB in APG chain
5C	A	4	IFCTPREV	Previous IFCB in APG chain
60	A	4	IFCTPDTP	Incore product descr tab
64	A	4	IFCTPAPG	Parent application group
68	A	4	IFCTFAPG	First application group
6C	F	4	IFCTTRCB	Private trace bits
	1...		IFCT5080	0 5080IO for PHIGS/5080
6C	B	31 bits	IFCTPRVT	Tested by trace
	.1..		IFCTPIM	1 Image manipulation
	..1.		IFCTPID	2 Image devices
	...1		IFCTPDS	3 Symbol sets
 1...		IFCTPDSO	4 Dsopen parameter tracing
1..		IFCTPGDP	5 FSM/GDP tracing
1.		IFCTPTXT	6 FSM/TEXT tracing
1		IFCTPCSF	7 Composite document structured fields
6D	1...		IFCTPHRG	8 HRIG data stream
	.1..		IFCTPVSE	9 VSE/Batch tracing
	..1.		IFCTPCLK	10 PCLink
	...1		IFCTPCGM	11 CGM report
 1111		*	
6E	1111 1111		*	
6F	1111 1111		*	
70	C	8	IFCTAIC0	AIC rcpcpnt = 00
70	A	4	IFCTAICE	AIC func processor entry point
74	A	4	IFCTAICS	AIC storage anchor
78	C	152	IFCTGEN	General module declarations
78	C	8	IFCTTSI0	TSI rcpcpnt = 04
78	C	1	IFCTTSID	TSI environment identifier
79	C	3	*	Reserved
7C	A	4	IFCTTSIS	TSI storage anchor

Offset	Type	Length	Name	Description
80	C	8	IFCTESI0	ESI rcpcpnt = 08
80	A	4	IFCTESIE	ESI entry point
84	A	4	IFCTESIS	ESI storage anchor
88	C	8	IFCTFSM0	FSM (GDDM) rcpcpnt = 0C
88	A	4	IFCTFSME	FSM entry point
8C	A	4	IFCTFSMS	FSM storage anchor
90	C	8	IFCTBGS0	BGS (bus/grphcs) rcpcpnt = 10
90	A	4	IFCTBGSE	ASC entry point
94	A	4	IFCTBGSS	ASC storage anchor
98	C	8	IFCTPLT0	PLT (plot util) rcpcpnt = 14
98	A	4	IFCTPLTE	PLT entry point
9C	A	4	IFCTPLTS	PLT storage anchor
A0	C	8	IFCTISE0	ISE (image editor) rcpcpnt = 18
A0	A	4	IFCTISEE	ISE entry point
A4	A	4	IFCTISES	ISE storage anchor
A8	C	8	IFCTCSI0	CSI (misc services) rcpcpnt = 1C
A8	A	4	IFCTCSIE	CSI entry point
AC	A	4	IFCTCSIS	CSI storage anchor
B0	C	8	IFCTVSE0	VSE (vector editor) rcpcpnt = 20
B0	A	4	IFCTVSEE	VSE entry point
B4	A	4	IFCTVSES	VSE storage anchor
B8	C	8	IFCTOUT0	OUT (print) rcpcpnt = 24
B8	A	4	IFCTOUTE	OUT entry point
BC	A	4	IFCTOUTS	OUT storage anchor
C0	C	8	IFCTIMS0	IMS (supervisor) rcpcpnt = 28
C0	A	4	IFCTIMSE	IMS entry point
C4	A	4	IFCTIMSS	IMS storage anchor
C8	C	8	IFCTNSR0	NSR (NUM support) rcpcpnt = 2C
C8	A	4	IFCTNSRE	NSR entry point
CC	A	4	IFCTNSRS	NSR storage anchor
D0	C	8	IFCTIMD0	IMD (map editor) rcpcpnt = 30
D0	A	4	IFCTIMDE	IMD entry point
D4	A	4	IFCTIMDS	IMD storage anchor
D8	C	8	IFCTFIG0	PHIGS mapper rcpcpnt = 34
D8	A	4	IFCTFIGE	PHIGS mapper entry point
DC	A	4	IFCTFIGS	PHIGS mapper storage anchor
E0	C	8	IFCTGKS0	GKS rcpcpnt = 38
E0	A	4	IFCTGKSE	GKS entry point
E4	A	4	IFCTGKSS	GKS storage anchor
E8	C	8	IFCTIMC0	IMC rcpcpnt = 3c
E8	A	4	IFCTIMCE	IMC entry point
EC	A	4	IFCTIMCS	IMC storage anchor
F0	C	8	IFCTCDU0	Composite Document Print Utility rcpcpnt = 40
F0	A	4	IFCTCDUE	CDPU entry point
F4	A	4	IFCTCDUS	CDPU storage anchor
F8	C	8	IFCTIVU0	Image View Utility rcpcpnt = 44
F8	A	4	IFCTIVUE	IVU entry point
FC	A	4	IFCTIVUS	IVU storage anchor
100	C	8	*	Reserved rcpcpnt = 48
100	A	4	*	
104	A	4	*	
108	C	8	*	Reserved rcpcpnt = 4c
108	A	4	*	
10C	A	4	*	
110	C	*	IFCBSSA	Subsystem save area based entry declarations
0	C	8	IFCTMODX	(15)

Offset	Type	Length	Name	Description
0	A	4	IFCTMODE	General module entry point
4	A	4	IFCTMODS	General module storage anchor

ADMTIFCE – GDDM interface control error feedback

Function

This control block contains error feedback fields, providing ancillary information relating to the response code in the Interface Control Block (IFCB). See “ADMTIFCB – interface control block (IFCB)” on page 159. The IFCE is normally addressed via a pointer in the Interface Control Table (IFCT).

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTIFCE	0	IFCEERRK	68	IFCES1	1C
IFCEA1	C	IFCEEXTN	4	IFCES2	26
IFCEA2	10	IFCEFC	4	IFCES3	30
IFCEA3	14	IFCEFLGS	8	IFCES4	3A
IFCEA4	18	IFCEL2	1C	IFCETXT	C
IFCEBLDF	8	IFCEPREF	8	IFCET1	14
IFCEC1	1C	IFCEPRET	9	IFCET2	18
IFCEC2	30	IFCEQQ	0	IFCEUSRF	8

Storage layout

Offset	Type	Length	Name	Description
0		108	ADMTIFCE	Interface error feedback block
0	B	4	IFCEQQ	Request code
4		8	IFCEEXTN	Extended error indicators. (The following fields are meaningful only if IFCBFC msg no.=X'FFF'.)
4	F	4	IFCEFC	Extended error msg no.
8	B	1	IFCEFLGS	Flag byte
1...			IFCEBLDF	Text already built
.1..			IFCEPREF	Non-GDDM prefix required (1=> Non-GDDM)
..1.			IFCEUSRF	Non-GDDM/IMD msg required (1=> Non-GDDM and non-IMD)
...1 1111				Spare
9	C	3	IFCEPRET	Required non-GDDM prefix
C	C	68	IFCETXT	Msg text excluding prefix, msg number, and severity, or, encoded message inserts
C	F	4	IFCEA1	Arith insert field 1
10	F	4	IFCEA2	Arith insert field 2
14	F	4	IFCEA3	Arith insert field 3
14	F	4	IFCET1	Text insert field 1
18	F	4	IFCEA4	Arith insert field 4
18	F	4	IFCET2	Text insert field 2
1C	C	40	IFCEL2	Character insert long field
1C	C	20	IFCEC1	Character insert field
1C	C	10	IFCES1	Short character insert field 1

Offset	Type	Length	Name	Description
26	C	10	IFCES2	Short character insert field 2
30	C	20	IFCEC2	Character insert field 2
30	C	10	IFCES3	Short character insert field 3
3A	C	10	IFCES4	Short character insert field 4
44	C	12	*	Spare
50	C	24	*	Spare
68	F	4	IFCEERRK	Key for error message service

ADMTNICK – nickname list entry format

Function

This control block defines the structure of a nickname list entry. The nickname lists are chained from field DFT0NICK in the general defaults table; see page 151.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTNICK	0	NICKFLGS	10	NICKSFAM	20
NICKAPPF	10	NICKFPTR	8	NICKSNMC	28
NICKBPTR	C	NICKHEAD	0	NICKSNMP	14
NICKDESC	40	NICKLENG	4	NICKTFAM	2C
NICKDESCF	10	NICKPRCC	3C	NICKTNMC	30
NICKDESP	20	NICKPRCP	1C	NICKTNMP	18
NICKDVTK	34	NICKSCNF	10		

Storage layout

Offset	Type	Length	Name	Description
0	C	*	ADMTNICK	Nickname list entry
0	C	4	NICKHEAD	'Nick' eye catcher
4	F	4	NICKLENG	Length of entry
8	A	4	NICKFPTR	Forward chain pointer
C	A	4	NICKBPTR	Back chain pointer
10	C	4	NICKFLGS	Flags
	1...		NICKSCNF	Used by FSM for scan control
	.1..		NICKAPPF	0 -> Replace option specified 1 -> append option (default)
	..1.		NICKDESCF	0 -> Desc is valid 1 -> Desc is invalid
14	A	4	NICKSNMP	Addr (source name data)
18	A	4	NICKTNMP	Addr (target name data)
1C	A	4	NICKPRCP	Addr (procopt data)
20	A	4	NICKDESP	Addr (desc data)
24	F	4	NICKSFAM	Source family
28	F	4	NICKSNMC	Count of source name dwords
2C	F	4	NICKTFAM	Target family
30	F	4	NICKTNMC	Count of target name dwords
34	C	8	NICKDVTK	Device token
3C	F	4	NICKPRCC	Count of procopt fwords
40	F	4	NICKDESC	Count of desc fwords
44	C	*	*	Variable length data

ADMTTCA – terminal services interface control area (TCA)

Function

This control block contains the addresses, data areas, and flags to be retained by the terminal service interface area (TSIA) across invocations. One instance of this control block occurs in each instance of the TSIA.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTTCA	0	TCALASTP	C	TCASTFIP	14
TCAACTVF	10	TCANEXTP	8	TCASTFOP	14
TCADUMLF	10	TCANOTF	10	TCASTFRL	14
TCADVTOK	1C	TCAQRYBF	10	TCASTFSL	14
TCAERRFC	16	TCAROPNF	10	TCASTFTL	14
TCAFLGCS	10	TCARROUP	18	TCASTGL	4
TCAFLGST	14	TCARRSVF	10	TCASYSL	28
TCAF2NCC	10	TCASSIA	0	TCASYSP	24
TCAID	0	TCASTFBS	14		

Storage layout

Offset	Type	Length	Name	Description
0	C	164	ADMTTCA	TSIA control area
0	C	84	TCASSIA	Header
0	C	4	TCAID	Eye-catcher TCA
4	F	4	TCASTGL	Length of acquired storage
8	A	4	TCANEXTP	Address (next TCA in the ring)
C	A	4	TCALASTP	Address (last TCA in the ring)
10	C	4	TCAFLGCS	Current status flags
	1... ..		TCANOTF	1 => Not first notable I/O
	.1.. ..		TCARRSVF	1 => Resource reserved via CSI
	..1.		TCAACTVF	1 => TSIA active
	...1		TCAROPNF	1 => Resource is open
 1...		TCADUMLF	1 => Dummy adapter is loaded
1..		TCAQRYBF	1 => I/P buffer is query reply
1.		TCAF2NCC	1 => Family 2 I/P with nospan, nocc
1		*	Reserved
	1111 1111		*	Reserved
	1111 1111		*	Reserved
	1111 1111		*	Reserved
14	C	2	TCAFLGST	State flags - valid next ops
	1... ..		TCASTFBS	Base functions, includes
				TQRY, RNIT, TERM, ACQ, O/P Buf
	.1.. ..		TCASTFOP	Output ops (Send/Converse)
	..1.		TCASTFIP	Input ops (Receive)
	...1		TCASTFRL	Release input buffer
 1...		TCASTFTL	Not first/only TMIT
1..		TCASTFSL	Not first/only segment
11		*	Reserved
	1111 1111		*	Reserved
16	F	2	TCAERRFC	Internal IFCBFC-type error code
18	A	4	TCARROUP	If dummy device, A(real router)
1C	C	8	TCADVTOK	Device token
24	A	4	TCASYSP	Address(TCA sysopt-list)
28	F	4	TCASYSL	Length(TCA sysopt-list)

Offset	Type	Length	Name	Description
2C	C	20	*	Spare fields
40	C	56	ADMTTRB	Terminal request block
78	C	44	ADMTTDB	Terminal descriptor block

ADMTTDB – terminal descriptor block (TDB)

Function

This control block contains fields and flags that describe the characteristics of the terminal. It includes the terminal type, its modes of operation, screen sizes, and link protocol.

It is a part of the terminal services interface control area (TCA). All offsets defined below are from the start of the TCA.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTRDBE	A4	TDBGAKAT	78	TDBREST	7B
ADMTTDB	78	TDBGAMON	78	TDBSCSBL	7A
TDBAPKB	7A	TDBHWS	7E	TDBSCSFB	7A
TDBAPLF	7A	TDBINPT	7C	TDBSDLBS	7D
TDBAPTU	7A	TDBINPUT	7C	TDBSDLC	7D
TDBAPTX	7A	TDBIO	7C	TDBSVBSZ	8A
TDBASCDR	7C	TDBIPDS	98	TDBSZ	7E
TDBASZ	82	TDBIRPI	9A	TBDSZAL	79
TDBASZC	84	TDBISTM	98	TDBSZF	79
TDBASZR	82	TDBLCLRM	7D	TDBSZMD	79
TDBAUXON	78	TDBLINKF	7D	TDBTEKGL	7C
TDBBSC	7D	TDBLOCAL	7D	TDBTEKPX	90
TDBCOLMP	94	TDBLU1P	78	TDBTEKPY	92
TDBCURBO	7B	TDBLU3P	78	TDBTRBSZ	8C
TDBDECGL	7C	TDBMISCF	7B	TDBTXKB	7A
TDBDEVAD	9C	TDBMOVEF	7B	TDBTXPR	7A
TBDFDB	78	TDBMPAGE	86	TDBTYPE	78
TBDDUMMY	7C	TDBMPCOL	88	TDBUNF	78
TDBFLAGS	78	TDBMPROW	86	TDBUSESF	7B
TDBFSSAV	8A	TDBMSEG	7B	TDBWSF	7B
TDBF4DEP	86	TDBMXMT	7B	TDB3270E	7B
TDBF4LNW	8A	TDBOUTPT	7C	TDB3277	78
TDBF4PIX	90	TDBPSZ	7E	TDB3278	78
TDBF4PIY	92	TDBPSZC	80	TDB38XX	78
TDBF4UNI	8E	TDBPSZR	7E	TDB3800	78
TDBF4WID	88	TDBREMOT	7D	TDB4250	78

Storage layout

Offset	Type	Length	Name	Description
78	C	44	ADMTTDB	Terminal descriptor block
78	B	6	TDBFLAGS	
78	B	1	TDBTYPE	Terminal type flags
	1...		TDB3277	Old type (3271,72,77,84,88)
	1...		TDB3800	3800 (Sonora) printer type
	.1..		TDB3278	New type (3276,78,87,89)

Offset	Type	Length	Name	Description
	.1..		TDB4250	4250 (BREG) printer type
	.1..		TDBLU1P	Device is LU1 (SCS) printer
	.1..		TDB38XX	38XX () printer type
	...1		TDBLU3P	Device is LU3 printer
	...1		TDBUNF	Unformatted HRIG output
	... 1...		TDBGAMON	Graphics attachment printer
1..		TDBGAKAT	Graphics attachment (Katakana) (Set in addition to TDBGAMON)
1.		TDBAUXON	Auxiliary only device
1		TDBDFDB	1 => This is 'DFT' terminal on subsystem that supports line O/P DBCS
79	B	1	TDBSZF	Terminal size flags
	1...		TDBSZMD	Screen size is modifiable
	.1..		TDBSZAL	Current screen if TDBSZMD = 0 1=> Alternate, 0 => Primary
	..11 1111		*	reserved
7A	B	1	TDBAPLF	APL/Text flags
	1...		TDBAPTU	APL Text FEA status unknown
	.1..		TDBAPT X	APL Text FEA present
	.1..		TDBTXKB	Text keyboard known
	...1		TDBAPKB	APL Keyboard known
	... 1...		TDBTXPR	Textprint feature on
1..		TDBSCSFB	0 = SCS base codes (NL, FF) supported (LU 1) 1 = SCS full base codes (NL, FF, BS, CR, LF) supported (LU 1)
1.		TDBSCSBL	1 = SCS BEL code supported (LU 1)
1		*	Reserved
7B	B	1	TDBMISCF	Miscellaneous flags
	1...		TDBCURBO	Cursor field return type
	.1..		TDBMSEG	0=Single segm,1=Multi segm xmit
	.1..		TDBMXMT	0=Single xmit,1=multi xmit xmit
	...1		TDBREST	1=Perishable ps stores
	... 1...		TDB3270E	0=Do not use 3270E outgoing
1..		TDBWSF	1=This box supports WSF (LU 2/LU 3) or FM1 for LU 1
1.		TDBUSESF	1=Always use structured fields if supported by box
1		TDBMOVEF	Mode of adapter operation
7C	B	1	TDBIO	I/O flags
	1...		TDBINPT	Adapter will return input
	.1..		TDBOUTPT	Adapter will accept output
	.1..		TDBDUMMY	Device is a dummy device
	...1		TDBINPUT	Input path to dev available
	... 1...		TDBDECGL	Ascii graphics DEC device
1..		TDBTEKGL	Ascii graphics TEK device
11		TDBASCDR	Ascii graphics device level
7D	B	1	TDBLINKF	Type of communications link
	11..		TDBLCLRM	These bits have values: (11 = Invalid 10 = Device is local 01 = Device is remote 00 = Type is unknown)
	1...		TDBLOCAL	Device is locally ATT
	.1..		TDBREMOT	Device is remotely at

Offset	Type	Length	Name	Description
	..11		TDBSDLBS	These bits have values: (11 = Invalid 10 = Link is SDLC 01 = Link is BSC 00 = Type is unknown)
	..1.		TDBSDLC	Link is SDLC
	...1		TDBBSC	Link is bisync
 1111		*	Reserved
7E	C	22	TDBHWS	
7E	C	8	TDBSZ	Screen sizes
7E	C	4	TDBPSZ	Primary screen sizes
7E	F	2	TDBPSZR	Primary screen size (row)
80	F	2	TDBPSZC	Primary screen size (cols)
82	C	4	TDBASZ	Alternate screen sizes
82	F	2	TDBASZR	Alternate screen size (rows)
84	F	2	TDBASZC	Alternate screen size (cols)
86	C	4	TDBMPAGE	Maximum page size
86	F	2	TDBMPROW	Maximum no. of rows
86	F	2	TDBF4DEP	Family 4 medium depth
88	F	2	TDBMPCOL	Maximum no. of columns
88	F	2	TDBF4WID	Family 4 medium width
8A	C	2	TDBFSSAV	
8A	F	2	TDBSVBSZ	FSSAVE buffer size
8A	F	2	TDBF4LNW	Family 4 pels/unit linewidth
8C	F	2	TDBTRBSZ	Transmission buffer size
8E	F	2	TDBF4UNI	Family 4 medium units
90	F	2	TDBF4PIX	Family 4 pels per inch x
90	F	2	TDBTEKPY	Ascii graphics screen width
92	F	2	TDBF4PIY	Family 4 pels per inch y
92	F	2	TDBTEKPY	Ascii graphics screen depth
94	A	4	TDBCOLMP	To color master entry
98	C	4	TDBIPDS	
98	F	2	TDBISTM	Length of STM query reply
9A	F	2	TDBIRPI	Length of RPI query reply
9C	F	4	TDBDEVAD	Device address
A0	F	4	*	Spare fields

ADMTTRB – terminal request block (TRB)

Function

This control block contains fields and flags for handling I/O operations with the device. It contains various buffer addresses and lengths, and fields that show the state of the I/O.

It is a part of the Terminal Services Interface Control Area (TCA). All offsets defined below are from the start of the TCA.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTTRB	40	TRBINLH	56	TRBOSTPS	5B
ADMTTRBE	78	TRBINP	50	TRBOSTRS	5B
TRBCMPNT	60	TRBLROUP	40	TRBOSTSC	5B
TRBEFLG1	64	TRBOCD	59	TRBOSYS	5A
TRBEFLG2	65	TRBOCDOB	59	TRBOTDB	5A
TRBEFLG3	66	TRBODEV	5A	TRBOUTL	4C
TRBEFLG4	67	TRBODR	59	TRBOUTLH	4E
TRBERROR	64	TRBOEB	59	TRBOUTP	48
TRBERS	64	TRBOFLG1	58	TRBRQUEST	58
TRBERSIO	64	TRBOFLG2	59	TRBRSPND	5C
TRBERSPS	64	TRBOFLG3	5A	TRBSESS1	68
TRBERSSC	64	TRBOFLG4	5B	TRBSLINK	68
TRBERS1	64	TRBOFMH	59	TRBSLKBY	68
TRBEWBB	74	TRBOFMHI	59	TRBSRCV1	68
TRBEWBBF	74	TRBOFSTS	58	TRBSRFL1	68
TRBICD	5D	TRBOFSTX	58	TRBTCAP	44
TRBIEB	5D	TRBOLSTS	58	TRBUACK	6C
TRBIFLG1	5C	TRBOLSTX	58	TRBUACKL	6C
TRBIFLG2	5D	TRBOMIDS	58	TRBUACKP	70
TRBIFLG3	5E	TRBOMIDX	58		
TRBIFLG4	5F	TRBOPART	59		
TRBIFMH	5D	TRBOQUERY	5A		
TRBIFSTS	5C	TRBORDIM	59		
TRBIFSTX	5C	TRBORQNN	5B		
TRBILSTS	5C	TRBORQPS	5B		
TRBILSTX	5C	TRBORQRS	5B		
TRBIMIDS	5C	TRBORQSC	5B		
TRBIMIDX	5C	TRBORQSY	5B		
TRBINL	54	TRBOSTNN	5B		

Storage layout

Offset	Type	Length	Name	Description
40	C	56	ADMTTRB	Terminal request block
40	A	4	TRBLROUP	Address of TSI router module
44	A	4	TRBTCAP	Address of TSI anchor block
48	A	4	TRBOUTP	Address of output buffer
4C	F	4	TRBOUTL	Length of output segment
4C	F	2	*	First h/word
4E	F	2	TRBOUTLH	Length as h/word (for CICS)
50	A	4	TRBINP	Address of input segment
54	F	4	TRBINL	Length of input segment
54	F	2	*	First h/word
56	F	2	TRBINLH	Length as h/word (for CICS)
58	C	4	TRBRQUEST	TRB request flags for call options
58	B	1	TRBOFLG1	Segment and transmit sequence
	1...		TRBOFSTX	Transmit(first)
	.1..		TRBOMIDX	Transmit(middle)
	..1.		TRBOLSTX	Transmit(last)
	...1		*	Reserved
 1...		TRBOFSTS	Segment(first)
1..		TRBOMIDS	Segment(middle)
1.		TRBOLSTS	Segment(last)
1		*	Reserved
59	B	1	TRBOFLG2	Session control, FMH and wait code

Offset	Type	Length	Name	Description
	1...		TRBOFMH	FMH required
	.1..		TRBOCD	Return flow with change direction (i.e. transmission is part of a converse call)
	..1.		TRBOEB	Return flow with end bracket
	...1		TRBORDIM	Return immediately if no IN
 1...		TRBOCDOB	Return flow with change direction on outbound transmission (as architecturally required by outbound data streams containing "read" or "query" commands)
1..		TRBOPART	Screen is being operated using partition(s)
1.		TRBOFMHI	FMH already included
1		TRBODR	SNA definite response
5A	B	1	TRBOFLG3	ADMMTQRY option flags
	1...		TRBOQUERY	Return query reply
	.1..		TRBOTDB	Return terminal descriptor block
	..1.		TRBODEV	Return device options
	...1		TRBOSYS	Return subsystem options
 1111		*	Reserved
5B	B	1	TRBOFLG4	Resource usage flags
	1111		TRBORQRS	Resources required by current transmission. '0000'b implies default of (screen + ps)
	1...		TRBORQSC	Screen buffer required
	.1..		TRBORQPS	PS buffer required
	..1.		TRBORQSY	I/O sync reqd. to force TSO HOLD
	...1		TRBORQNN	Inhibits DFT inference
 1111		TRBOSTRS	Resources changed by current transmission. '0000'b implies default as follows, EW/EWA/W all except PS WSF ALL
 1...		TRBOSTSC	Screen buffer changed
1..		TRBOSTPS	PS buffer changed
1.		*	Reserved
1		TRBOSTNN	Inhibits DFT inference
5C	C	4	TRBRSPND	TRB response flags from adapter
5C	B	1	TRBIFLG1	Segment and transmit sequence
	1...		TRBIFSTX	Transmit(first)
	.1..		TRBIMIDX	Transmit(middle)
	..1.		TRBILSTX	Transmit(last)
	...1		*	Reserved
 1...		TRBIFSTS	Segment(first)
1..		TRBIMIDS	Segment(middle)
1.		TRBILSTS	Segment(last)
1		*	Reserved
5D	B	1	TRBIFLG2	Session state and fmh present
	1...		TRBIFMH	FMH present in data
	.1..		TRBICD	Returned flow with change direction
	..1.		TRBIEB	Returned flow with end bracket
	...1 1111		*	Reserved
5E	B	1	TRBIFLG3	Reserved
5F	B	1	TRBIFLG4	Reserved
60	F	4	TRBCMPNT	Component NDX to file set
64	C	4	TRBERROR	TRB error flags from adapter
64	B	1	TRBEFLG1	Required unavailable resources flags. if there is a reshew error, these flags identify which resources need reshew.
	1111		TRBERS	
	1...		TRBERSSC	Screen buffer
	.1..		TRBERSPS	PS buffer
	..1.		*	Reserved

Offset	Type	Length	Name	Description
	...1		TRBERSIO	Last transmission contents
 1111		TRBERS1	Reserved
65	B	1	TRBEFLG2	Reserved
66	B	1	TRBEFLG3	Reserved
67	B	1	TRBEFLG4	Reserved
68	C	1	TRBSESS1	Further session control flag
68	B	1	TRBSRFL1	Receive processing flags
	1...		TRBSRCV1	First receive processed (currently used only for CICS pseudo conv)
	.1..		TRBSLINK	Transmission for GDDM-PCLK or GDDM-OS/2 Link
	..1.		TRBSLKBY	Transmission for 'link' was bypassed
	...1 1111		*	Reserved
69	C	3	*	Spare
6C	C	8	TRBUACK	Unexpected acknowledgement
6C	F	2	TRBUACKL	ACK buffer length
6E	C	2	*	Reserved
70	A	4	TRBUACKP	-> FSM ACK buffer
74	C	1	TRBEWBB	Erase Write before buffer
	1...		TRBEWBBF	Erase Write transmission flag
	.111 1111		*	Reserved
75	C	3	*	Further spare fields
78	C	0	ADMTTRBE	End of request block

_____ End of Diagnosis, Modification or Tuning Information _____

Appendix B. Abend codes

This appendix lists in numeric order the abend codes produced by GDDM Base, GDDM-PGF, GDDM-IMD, GDDM-IVU, GDDM-GKS, GDDM-REXX, and GDDM-PCLK.

Each abend code is listed with information you may find helpful when looking for the cause of the abend. A code for the component from which the abend was issued, is given in the “Component” column; these codes are explained on page 39.

In some instances, the abend is preceded by the following message:

```
ADM0309 U GDDM ABEND n AT OFFSET X'xxxx' IN 'a'
```

where:

- n is the abend code.
- X'xxxx' is the offset of the location in the failing module where the abend was raised.
- a is the module “eye-catcher”, which consists of the module name, compile date, GDDM version, and (possibly) service level.

For all GDDM abends, general-purpose register 2 (GPR02) is set to the address of the interface control block (IFCB). IFCTEFBP in the IFCB is set to the address of the error feedback block (IFCE). Within the IFCE fields, IFCEA1 is set to the abend code, IFCEA2 is set to the offset, and IFCEL2 is set to the eye-catcher.

Under CICS, the first digit of the abend code is replaced by a “G” to identify the abend as originating from GDDM.

Under IMS, the codes listed have 1000 added before issuing an abend, to avoid conflict with standard IMS abend codes. Thus, listed abend code 1163 appears as G163 under CICS, and as 2163 under IMS.

Under VM, the abend code may appear in hexadecimal if the issuing module is in a saved segment.

Note: Under VM/SP, saved segments are called *discontiguous saved segments* (DCSS), while under VM/XA and VM/ESA they are called *named saved systems* (NSS).

Under VSE, the abend code appears in register 5 when a dump has been requested. It also appears as the return code from the job step.

GDDM-IVU abends appear in register 0 and result from the following main causes:

- An error record with severity greater than zero returned from GDDM Base.
- Nonzero return codes from the GDDM-IVU input data converter (ADM5CV).
- Nonzero return codes from the GDDM-IVU default value inserter (ADM5DF).
- Nonzero return codes from the GDDM-IVU name handler (ADM5NM).
- Nonzero return codes from the GDDM-IVU screen handler (ADM5SH).
- Errors detected during GDDM-IVU storage acquisition and release.

Other causes of GDDM-IVU abends are listed by the relevant module.

abends

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1000	03E8	ADME000C	ESI	Error log records present	1115	045B	ADMDCCD	FSM	Invalid conversion type
1050	041A	ADMACFP	AIC	Invalid request code	1120	0460	ADMDSOO	FSM	Invalid default POPCODE
1051	041B	ADMACUP ADMACLP	AIC	Abend on FSTERM requested	1121	0461	ADMDSOO	FSM	Invalid code parameter
1053	041D	ADME000I	AIC	AIC-detected FSINIT error	1122	0462	ADMDSOO	FSM	Invalid system options
1054	041E	ADME000I	AIC	AIC-detected non-FSINIT error	1123	0463	ADMDSOO	FSM	Input option length exceeds default
1058	0422	ADMACSSP	AIC	Invalid call to ADMACSSP	1124	0464	ADMDSMC	FSM	Invalid request code
1059	0423	ADMACDP	AIC	SOS stack overflow	1125	0465	ADMDSMC	FSM	Invalid control blocks
1060	0424	ADMACSSP ADMACDP	AIC	Overflow stack overflow	1126	0466	ADMDSOO	FSM	Defaults definition error
1062	0426	ADMACUPS	AIC	Local EUDS area too small	1127	0467	ADMDSF2 ADMDSF3 ADMDSF4	FSM	PROCOPT code error
1063	0427	ADMACUP ADMACUPS ADMATPE	AIC	ADME0DFx entry wrong (x = subsystem ID)	1128	0468	ADMDSA1 ADMDSF1 ADMDSF2 ADMDSF3 ADMDSF4 ADMDSFQ	FSM	PROCOPT number error
1064	0428	ADMACLP	AIC	Product information unavailable	1129	0469	ADMDOOC ADMDSFQ	FSM	No print control option
1100	044C	ADMDSRO ADMD3SXF	FSM	Invalid request code	1130	046A	ADMDOAL	FSM	ASQFLD error on primary device
1101	044D	ADMDSRO	FSM	Invalid FCA pointer	1133	046D	ADMDOCO	FSM	No primary device
1102	044E	ADMDSRO	FSM	Not FSM component code	1139	0473	ADMDOBS	FSM	Invalid code passed to ADMDOBS
1103	044F	ADMDSL	FSM	Invalid module group	1150	047E	ADMDSF1 ADMDSF2 ADMDSF3 ADMDSF4 ADMDSFQ	FSM	Invalid system pattern
1104	0450	ADMDSL	FSM	Invalid subcomponent code for module group	1151	047F	ADMDSF1 ADMDSF2 ADMDSF3 ADMDSF4 ADMDSF6	FSM	Temporary buffer overflow
1105	0451	ADMDSL	FSM	No DST address in DCA	1153	0481	ADMDSFQ	FSM	Error during regenerate
1106	0452	ADMDSDS	FSM	Invalid PROCOPT list	1154	0482	ADMDSF1 ADMDSF2 ADMDSF3 ADMDSF4 ADMDSF6	FSM	Invalid system symbol set
1107	0453	ADMDSDS	FSM	All device IDs in use	1155	0483	ADMDSF1 ADMDSF2 ADMDSF3 ADMDSF4 ADMDSF6	FSM	Invalid pointer
1108	0454	ADMDSF1 ADMDSOO	FSM	Invalid group length	1156	0484	ADMDSF1 ADMDSF2 ADMDSF3 ADMDSF4 ADMDSF6	FSM	SEG delete
1109	0455	ADMDSOO	FSM	Table size exceeded	1157	0485	ADMDSF1 ADMDSF2 ADMDSF3 ADMDSF4 ADMDSF6	FSM	Invalid family 1 symbol set
1110	0456	ADMDCDS	FSM	Unsupported RCP code					
1111	0457	ADMDCDS	FSM	Zero pointer detected					
1112	0458	ADMDCPC	FSM	No current page					
1114	045A	ADMDCGS	FSM	No SST entry for pattern or marker set (GSLSS or GSDSS)					

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1158	0486	ADMDGI1 ADMDGI3 ADMDGI4 ADMDGI5 ADMDGI6	FSM	Invalid 'function' code	1201	04B1	ADMDSSST	FSM	None of the 'types' match the device characteristics, so a default 'type' cannot be selected. Change ADMDATRN TRNUCTP is 0. No upper case translate table defined TRNPTP is 0. No protection translate table defined TRNTYPEP(N) is 0. There is no address for a type descriptor block
1159	0487	ADMDGPCX	FSM	Default symbol set not found	1203	04B3	ADMDSDS ADMDSRO	FSM	
1160	0488	ADMDTC ADMDTCON ADMDTPGD	FSM	IEEE488 or BP1 error response to transmission of plotter data stream	1204	04B4	ADMDSSST	FSM	
1161	0489	ADMDDIMF ADMDDIMG	FSM	Invalid parameter	1205	04B5	ADMDSSST	FSM	
1162	048A	ADMDDIMG	FSM	Invalid projection (Place_Pos Mix Mode) for conversion to IOCA	1206	04B6	ADMDSSST	FSM	
1163	048B	ADMDDUFL	FSM	Invalid load type	1207	04B7	ADMDSII	FSM	Number of types defined (TRNNTYPE) is zero or negative
1164	048C	ADMDDUFA ADMDWUFL ADMDXUFI ADMDXUFM	FSM	Invalid (that is, 0) pointer					
1165	048D	ADMDXUFM	FSM	Invalid function used					
1170	0492	ADMDBFN1	FSM	Invalid RCP code					
1171	0493	ADMDBFN2	FSM	Invalid RCP code					
1180	049C	ADMDEFN1	FSM	Invalid RCP code	1208	04B8	ADMDSUB	FSM	Invalid request parameter
1181	049D	ADMDEFN2	FSM	Invalid RCP code	1209	04B9	ADMDSIK	FSM	No alternate device QIB
1190	04A6	ADMD3DIF	FSM	Invalid device id	1211	04BB	ADMDASLC	FSM	Invalid request parameter
1191	04A7	ADMD3DQC ADMD3DQF	FSM	Array count mismatch	1230	04CE	ADMDHCO	FSM	Invalid parameter
1192	04A8	ADMD3DIF ADMD3DPG	FSM	Unsupported RCP code	1231	04CF	ADMDHIN	FSM	Invalid order code detected
1193	04A9	ADMD3DIS ADMD3DPL ADMD3EMU	FSM	Unsupported function code	1232	04D0	ADMDHIN	FSM	Invalid device cell size
1194	04AA	ADMD3FPI	FSM	Invalid projection for conversion to IOCA	1237	04D5	ADMDHAL ADMDHPA	FSM	Not enough system linetypes
1195	04AB	ADMD3DIF	FSM	ISPUT invalid span_state	1238	04D6	ADMDHUW	FSM	Invalid system linetype
1200	04B0	ADMDSSST	FSM	An invalid value of TRNDEFRT has been found in ADMDATRN. Change ADMDATRN	1239	04D7	ADMDHUA ADMDHUW	FSM	One-cell buffer control error
					1260	04EC	ADMDIHD	FSM	Area shading routing error
					1261	04ED	ADMDIHD	FSM	Error from FSPCRT for create default page
					1262	04EE	ADMDIHD	FSM	Error from FSPQRY for query default page
									Error from ASDFLD for define header line 1

abends

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1263	04EF	ADMDIHD	FSM	Error from ASTYPE for device	1302	0516	ADMEGFAC	ESI	Invalid RCP function code
1264	04F0	ADMDIHD	FSM	Error from ASCPUT for header data line 1			ADMEGFBC		
1265	04F1	ADMDIOC	FSM	Invalid PROCOPT list			ADMEGFDO		
							ADMEGFEB		
							ADMEGFEC		
1267	04F3	ADMDIPA	FSM	Error from FSPQRY for print page			ADMEGFEO		
							ADMEGFEP		
							ADMEGFES		
1268	04F4	ADMDIPA	FSM	Error from FSPDEL for print page			ADMEGFES		
1269	04F5	ADMDIGR	FSM	Old format print file – not supported			ADMEGFES		
					ADMEGFES				
					ADMEGFES				
1276	04FC	ADMDICO	FSM	No primary device	ADMEGFES				
1293	050D	ADMDGGP	FSM	Area greater than 60k	ADMEGFES				
1300	0514	ADMDGGP	ESI	Invalid ECA	ADMEGFES				
		ADMUPCT			ADMEGFES				
		ADMUPCV			ADMEGFES				
		ADMUPGT			ADMEGFES				
		ADMUPGV			ADMEGFES				
		ADMEIN0C			ADMEGFES				
		ADMEIN0D			ADMEGFES				
		ADMEIN0O			ADMEGFES				
		ADMEIN0V			ADMEGFES				
		ADMEROU			ADMEGFES				
1301	0515	ADMEROU	ESI	Invalid RCP subcomponent code	ADMEGFES				
		ADMEROU			ADMEGFES				
		ADMEROU			ADMEGFES				
		ADMEROU			ADMEGFES				
		ADMEROU			ADMEGFES				
		ADMEROU			ADMEGFES				
		ADMEROU			ADMEGFES				
		ADMEROU			ADMEGFES				
		ADMEROU			ADMEGFES				
		ADMEROU			ADMEGFES				
1303	0517	ADMECM0C	ESI	Function not supported	ADMECM0C				
		ADMECM0D			ADMECM0D				
		ADMECM0I			ADMECM0I				
		ADMECM0O			ADMECM0O				
		ADMECM0V			ADMECM0V				
		ADMEDSSV			ADMEDSSV				
		ADMEGFQD			ADMEGFQD				
		ADMEPCBI			ADMEPCBI				
		ADMEPRGI			ADMEPRGI				
		ADMEPROC			ADMEPROC				
1308	051C	ADMEPR0D	ESI	Program load failure – only if unconditional request	ADMEPR0D				
		ADMEPR0O			ADMEPR0O				
		ADMEPR0V			ADMEPR0V				
		ADMEPR0I			ADMEPR0I				
		ADMEPR0C			ADMEPR0C				
1309	051D	ADMECM0C	ESI	X-INST table already loaded or unloaded	ADMECM0C				
		ADMECM0D			ADMECM0D				
		ADMECM0I			ADMECM0I				
		ADMECM0O			ADMECM0O				
		ADMECM0V			ADMECM0V				
1310	051E	ADMECM0V	ESI	Nonzero return code from CMS SUBCOM					

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1311	051F	ADMECM0V	ESI	Nonzero return code from CMS DMSFRT	1317	0525	ADMEGFAC ADMEGFBC ADMEGFBCO ADMEGFDO ADMEGFEBV ADMEGFFFC ADMEGFGO ADMEGFII ADMEGFJI ADMEGFKV ADMEGFLO ADMEGFOD ADMEGFDP ADMEGFQD ADMEGFSD ADMEGF10 ADMEGF11 ADMEGF12 ADMEGF13 ADMEGF14 ADMEGF20 ADMEGF30 ADMEGF40 ADMEGF50 ADMEGF60 ADMEGF70 ADMEGF80 ADMEGF90	ESI	Invalid ORB or ORB parameters
1312	0520	ADMEROUI ADMERO0D ADMERO0O ADMERO0V ADMKSRVC	ESI	Unconditional GETMAIN failure					
1313	0521	ADMERO0D ADMERO0O ADMERO0V	ESI	FREEMAIN failure					
1314	0522	ADMEROUI ADMERO0C ADMERO0D ADMERO0O ADMERO0V	ESI	Request exceeds maximum storage					
1315	0523	ADMEGF00	ESI	Invalid family code					
1316	0524	ADMEGFAC ADMEGFBC ADMEGFBCO ADMEGFDO ADMEGFEBV ADMEGFFFC ADMEGFGO ADMEGFII ADMEGFJI ADMEGFKV ADMEGFLO ADMEGFOD ADMEGFDP ADMEGFQD ADMEGFSD ADMEGF10 ADMEGF11 ADMEGF12 ADMEGF13 ADMEGF14 ADMEGF20 ADMEGF30 ADMEGF40 ADMEGF50 ADMEGF60 ADMEGF70 ADMEGF80 ADMEGF90	ESI	Invalid FRB or FRB parameters	1318	0526	ADMEGF10 ADMEGF11 ADMEGF12 ADMEGF13 ADMEGF14 ADMEGF20 ADMEGF30 ADMEGF70 ADMEGF80 ADMEGF90	ESI	Invalid IFCT TSID field
					1319	0527	ADMEDCKI	ESI	Invalid address or length parameters
					1321	0529	ADMEGF40	ESI	Invalid piece-long parameter
					1322	052A	ADMEGFLO ADMEGFOD	ESI	Error from VSAM GENCB or MODCB
					1323	052B	ADMEPQ0O	ESI	Print queue not open
					1324	052C	ADMEPQ0O	ESI	Queue identifier changed
					1326	052E	ADMEOSLC ADMEOSLD ADMEOSLO ADMEOSQI	ESI	Invalid object type
					1327	052F	ADMEOSD0	ESI	Invalid name count type
					1328	0530	ADMEOSD0	ESI	Object not on queue
					1329	0531	ADMEGF30	ESI	Invalid object type

abends

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1330	0532	ADMESSM0	ESI	Invalid SSM request	1402	057A	ADMLFM10 ADMLRO00 ADMLRO1C ADMLRO1I ADMLRO1O ADMLRO1T ADMLRO1V ADMLRO2C ADMLRO2I ADMLRO2O ADMLRO2T ADMLRO2V ADMLRO3C ADMLRO3I ADMLRO3T ADMLRO3V ADMLRO4D ADMLRO4T ADMLRO4V	TSI	Invalid RCP code
1331	0533	ADMESSM0	ESI	Overlap found on FAQE					
1334	0536	ADMEGFCO ADMEGFGO ADMEPQ0O ADMEPR0O ADMERO0O	ESI	Invalid feedback or return from task switch exit					
1350	0546	ADMEINTI	ESI	ASMTDLI not link-edited					
1353	0549	ADMEPCBI	ESI	Invalid ordinal					
1354	054A	ADMEPCBI	ESI	Invalid PCB type					
1355	054B	ADMEDLII	ESI	Probable GDDM logic error in DL/I operation					
1356	054C	ADMEABNI ADMKSRVC	ESI	DL/I terminating condition found in scheduler subtask	1403	057B	ADMLCN1C ADMLFM10 ADMLRO00 ADMLSN1C ADMLSN1I ADMLSN1O ADMLSN1T ADMLSN1V ADMLSN20 ADMLSN30 ADMLSN40	TSI	Invalid send parameters
1380	0564	ADMEPRGI ADMEPR0C ADMEPR0D ADMEPR0O ADMEPR0V	ESI	Wrong program load form					
1381	0565	ADMEPRGI ADMEPR0C ADMEPR0D ADMEPR0O	ESI	Module loaded above 16 megabytes when GDDM initialized in 24-bit mode - second occurrence - possible recursion	1405	057D	ADMLQU1C ADMLQU1I ADMLQU1O ADMLQU1T ADMLQU1V	TSI	Abend after QDV requested via trace qualifier
1400	0578	ADMLFM10 ADMLRO00	TSI	TSIA already initialized	1406	057E	ADMLBC1V ADMLSN1V	TSI	Invalid command code
1401	0579	ADMLFM10 ADMLRO00 ADMLRO1C ADMLRO1I ADMLRO1O ADMLRO1T ADMLRO1V ADMLRO2C ADMLRO2I ADMLRO2O ADMLRO2T ADMLRO2V ADMLRO3C ADMLRO3I ADMLRO3T ADMLRO3V ADMLRO4D ADMLRO4T ADMLRO4V	TSI	TSIA not initialized or bad TCA					

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1407	057F	ADMLAC1C	TSI	Invalid TSIA state	1413	0585	ADMLIN1C	TSI	Invalid syslist parameters
		ADMLAC1I					ADMLIN1I		
		ADMLAC1O					ADMLIN1O		
		ADMLAC1T					ADMLIN1T		
		ADMLAC1V					ADMLIN1V		
		ADMLAC30					ADMLIN2C		
		ADMLAC40					ADMLIN2I		
		ADMLCN1C					ADMLIN2T		
		ADMLCN1O					ADMLIN2V		
		ADMLCN1T					ADMLIN3C		
		ADMLCN1V					ADMLIN3I		
		ADMLFM10					ADMLIN3T		
		ADMLRC1C					ADMLIN3V		
		ADMLRC1I					ADMLIN4D		
		ADMLRC1O					ADMLIN4T		
		ADMLRC1T			1420	058C	ADMLIN4V		
		ADMLRC1V					ADMLIN1T	TSI	Failure in initialize, TCLEARQ
		ADMLRC20			1422	058E	ADMLIN1T	TSI	Failure in initialize, STTMPMD
		ADMLRL1C					ADMLRC1T	TSI	Failure in receive, TGET
		ADMLRL1I			1423	058F	ADMLRC1T	TSI	Failure in receive, TCLEARQ 2
		ADMLRL1O					ADMLRN1I	TSI	Failure in send, STFSMODE
		ADMLRL1T			1425	0591	ADMLRN2I	TSI	Failure in send, TPUT
		ADMLRL1V					ADMLRN2O	TSI	Failure in send, TGET
		ADMLRL30			1427	0593	ADMLRN2O	TSI	Failure in send, TCLEARQ 2
		ADMLRL40					ADMLRN2O	TSI	Failure in send, STFSMODE
		ADMLRN1I			1428	0594	ADMLRN2O	TSI	Failure in send, STFSMODE
		ADMLRN2I					ADMLRN2O	TSI	Failure in send, STFSMODE
		ADMLRO00			1429	0595	ADMLRN2O	TSI	Failure in send, TPUT
		ADMLSN1C					ADMLRN2O	TSI	Failure in send, TGET
		ADMLSN1I			1430	0596	ADMLRN2O	TSI	Failure in send, TCLEARQ
		ADMLSN1O					ADMLRN2O	TSI	Failure in send, TCLEARQ 2
		ADMLSN1T			1431	0597	ADMLRN2O	TSI	Failure in send, TCLEARQ
		ADMLSN1V					ADMLRN2O	TSI	Failure in send, TCLEARQ 2
		ADMLSN20			1432	0598	ADMLRN2O	TSI	Failure in send, TCLEARQ
		ADMLSN30					ADMLRN2O	TSI	Failure in send, TCLEARQ 2
		ADMLSN40			1437	059D	ADMLRN2O	TSI	Failure in send, TCLEARQ
		ADMLTM1I					ADMLRN2O	TSI	Failure in send, TCLEARQ 2
1408	0580	ADMLAC1I	TSI	Invalid buffer parameters	1438	059E	ADMLRN2O	TSI	Failure in send, TCLEARQ
		ADMLRL1I					ADMLRN2O		
		ADMLRL1O					ADMLRN2O		
		ADMLRL1T					ADMLRN2O		
1410	0582	ADMLRL1V	TSI	Invalid CNVS parameters (TRB)	1439	059F	ADMLRN2O	TSI	Failure in send, TCLEARQ
		ADMLCN1C					ADMLRN2O		
		ADMLCN1I					ADMLRN2O		
		ADMLCN1O					ADMLRN2O		
1411	0583	ADMLCN1T	TSI	Invalid initialization parameters	1440	05A0	ADMLRN2O	TSI	Failure in send, TCLEARQ
		ADMLFM10					ADMLRN2O		
		ADMLRC1O					ADMLRN2O		
		ADMLRO00					ADMLRN2O		
1412	0584	ADMLIN1O	TSI	Invalid request parameters	1441	05A1	ADMLRN2O	TSI	Failure in send, TCLEARQ
		ADMLIN2O					ADMLRN2O		
		ADMLRN2O					ADMLRN2O		
		ADMLRN2O					ADMLRN2O		
1413	0585	ADMLRN2O	TSI	Invalid request parameters	1442	05A2	ADMLRN2O	TSI	Failure in send, TCLEARQ
		ADMLRN2O					ADMLRN2O		
		ADMLRN2O					ADMLRN2O		
		ADMLRN2O					ADMLRN2O		

abends

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1443	05A3	ADMLRL1T	TSI	Failure in release buffer, TCLEARQ	1529	05F9	ADM5EDT	IVU	File closure error
1450	05AA	ADMLIN1O	TSI	ICB invalid or not defined	1530	05FA	ADM5ERR	IVU	
1451	05AB	ADMLER1O	TSI	Failure in CB-type macro	1531	05FB	ADM5EX	IVU	
		ADMLIN1O			1532	05FC	ADM5EXP	IVU	
		ADMLQU1O			1533	05FD	ADM5IM	IVU	File closure error
		ADMLSN1O			1535	05FF	ADM5IM1	IVU	
		ADMLXR1O			1536	0600	ADM5INIT	IVU	
		ADMLTM1O			1537	0601	ADM5IP	IVU	
1452	05AC	ADMLTM1O	TSI	Session in use at termination	1538	0602	ADM5IV	IVU	File closure error
1461	05B5	ADMLER1V	TSI	Unexpected ADMLIO1V return code detected by ADMLER1V	1540	0604	ADM5LD	IVU	
					1542	0606	ADM5LDR	IVU	
					1543	0607	ADM5MAIN	IVU	
1462	05B6	ADMLTM2V	TSI	CSI resource list not empty after INVKOPUV	1545	0609	ADM5NMA	IVU	File closure error
1470	05BE	ADMLCN1C	TSI	TERMEPR return from CICS	1546	060A	ADM5OP	IVU	
1500	05DC	ADMBBGS	PGR	Invalid component code	1547	060B	ADM5PFK	IVU	
1501	05DD	ADMBBGS	PGR	Invalid BGDA pointer or overwritten (after initializing)	1548	060C	ADM5PFS	IVU	
1502	05DE	ADMBBGS	PGR	Invalid BGCB pointer or overwritten (after initializing)	1549	060D	ADM5PJ	IVU	File closure error
1503	05DF	ADMBBGS	PGR	Invalid subcomponent code (before searching branch table)	1550	060E	ADM5PJA	IVU	
1504	05E0	ADMBBGS	PGR	Invalid function	1551	060F	ADM5PJC	IVU	
		ADMBGCHR			1552	0610	ADM5PJL	IVU	
		ADMBGFI			1553	0611	ADM5PJS	IVU	File closure error
		ADMBGFLT			1554	0612	ADM5PJ1	IVU	
		ADMBGINT			1555	0613	ADM5PJ2	IVU	
		ADMBSET			1556	0614	ADM5PJ3	IVU	
1505	05E1	ADMBDRAX	PGR	Zero axis range	1557	0615	ADM5PR	IVU	File closure error
1506	05E2	ADMBDRAX	PGR	Zero tick interval	1558	0616	ADM5PR4	IVU	
1507	05E3	ADMBDRAX	PGR	Autotick leading digit calculation error	1559	0617	ADM5SC	IVU	
1520	05F0	ADM5CUR	IVU	Invalid input parameters	1560	0618	ADM5SCP	IVU	
1522	05F2	ADM5DC	IVU		1561	0619	ADM5SH	IVU	Invalid interrupt type
1524	05F4	ADM5ED	IVU		1562	061A	ADM5SHN	IVU	
1525	05F5	ADM5EDC	IVU		1563	061B	ADM5SI	IVU	
1526	05F6	ADM5EDD	IVU		1565	061D	ADM5VW	IVU	
1527	05F7	ADM5EDE	IVU		1566	061E	ADM5PJD	IVU	Invalid GDDM request
1528	05F8	ADM5EDF	IVU		1600	0640	ADMVHELP	VSE	
					1601	0641	ADMVFSM	VSE	
					1602	0642	ADMVEDRW	VSE	
							ADMVFSM		Nonzero feedback code from FSM
					1603	0643	ADMVHELP	VSE	
					1604	0644	ADMVHELP	VSE	
					1605	0645	ADMVUPDC	VSE	
					1606	0646	ADMVGRIN	VSE	Invalid order in help text
					1607	0647	ADMVGRIN	VSE	
					1686	0696	ADMPIGUS	ICU	
					1687	0697	ADMPDRAS	ICU	
					1688	0698	ADMPHWSH	ICU	Invalid PFKEY interrupt

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1689	0699	ADMPSUB0	ICU	Function error (abend subcode in register 0)	1708	06AC	ADMPAREC	ICU	Expected record not found (that is, no create and no return code)
1690	069A	ADMPDRSS	ICU	Symbol set error (abend subcode in register 0)	1709	06AD	ADMPAREC	ICU	Invalid record type
1691	069B	ADMPHWSH	ICU	Bad hierarchy – chart type unknown	1710	06AE	ADMPCONV ADMJCONV ADMDCGNV ADMDCRCNV	ICU GKS FSM FSM	Number conversion error found and no return code
1692	069C	ADMPDRAW	ICU	Chart-draw error (abend subcode in register 0)	1711	06AF	ADMPINFO	ICU	requested Function text not found
1693	069D	ADMPMAIN	ICU	No CHARTIDs	1712	06B0	ADMPWTXT	ICU	Record type not suitable for text display
1694	069E	ADMPRMLT	ICU	Screen formatting error	1713	06B1	ADMPLCMD	ICU	invalid row column
1695	069F	ADMPWMLT	ICU	Screen formatting error (abend subcode in register 0)	1714	06B2	ADMPMENU	ICU	Conversion type invalid for display of stored value
1696	06A0	ADMPMENW	ICU	Invalid call (including word not found)	1715	06B3	ADMPLCMD	ICU	invalid command
1697	06A1	ADMPBCHC	ICU	Invalid values table	1716	06B4	ADMPRCMD	ICU	Invalid command specified
1698	06A2	ADMPBIFC ADMPBCHC ADMPBDAT ADMPMAIN	ICU	Invalid RCP	1717	06B5	ADMPERR	ICU	Field in error does not exist
1699	06A3	ADMPIGDA	ICU	I/G drawing aid error (abend subcode in register 0)	1718	06B6	ADMPRECX	ICU	invalid command
1700	06A4	ADMPMENU	ICU	Menu index entry not found	1720	06B8	ADMPTXTL	ICU	Invalid record type for insert/delete
1701	06A5	ADMPFUNC ADMPIDIF	ICU	Invalid function request	1722	06BA	ADMPLSRT	ICU	invalid sort
1702	06A6	ADMPINFO	ICU	Not enough room in PF key or command information field to contain display information	1723	06BB	ADMPTERM	ICU	Termination error (abend subcode in register 0)
1703	06A7	ADMPHELP	ICU	Help panel index entry not found	1724	06BC	ADMPFUNC	ICU	No help panel to unload
1704	06A8	ADMPGET	ICU	Invalid op-code	1725	06BD	ADMPERR	ICU	Length of error text exceeds maximum (as shown by QA0ERRTL)
1705	06A9	ADMPSAVE	ICU	Invalid op-code	1726	06BE	ADMPRECX	ICU	Invalid record type for insert/delete
1707	06AB	ADMPCHEK ADMPRIV	ICU	Input field too long. Maximum is 132	1727	06BF	ADMPDRAN	ICU	Chart-note error (abend subcode in register 0)
					1728	06C0	ADMPINIT	ICU	Invalid DSQDEV call for alternate device

abends

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1729	06C1	ADMPINIT	ICU	Invalid DSQDEV call for primary device	1753	06D9	ADMOQPU	OPU	An invalid PROCOPT or PROCOPT value is contained within the header of the ADMPRINT file that was processed.
1731	06C3	ADMPBNUM ADMPWDIR	ICU	Unexpected type code in object descriptor	1757	06DD	ADMOPMT ADMOPUJ	OPU	Failure in SPINIT call
1732	06C4	ADMPWDIR	ICU	Internal ICU type code not found	1759	06DF	ADMOPUI	OPU	Unable to initialize GDDM
1733	06C5	ADMPWDIR ADMPWPS	ICU	ASQFLD error (column heading field ID not known)	1760	06E0	ADMYRO00	CSI	Invalid RCP subcomponent code
1734	06C6	ADMPWDIR ADMPWPS	ICU	ASRFMT error	1762	06E2	ADMYGQC	CSI	Invalid QUICKSELL request
1735	06C7	ADMPWDIR ADMPWPS	ICU	ASCPUT error	1763	06E3	ADMYGQC	CSI	Zero cellsize
1736	06C8	ADMPWATT	ICU	ASQFLD error (column heading field ID not known)	1764	06E4	ADMYASRT	CSI	Invalid argument on call to ADMTASRT
1737	06C9	ADMPDEEX	ICU	invalid command	1770	06EA	ADMKSCHD	IMS	Error from GU to I/O PCB
1738	06CA	ADMPRECT	ICU	invalid record type	1771	06EB	ADMKEXST	IMS	Input message exceeds input area size
1739	06CB	ADMPLSRT	ICU	item too long to be sorted	1772	06EC	ADMKTSWI	IMS	No CWA in SSAI
1740	06CC	ADMPWCMD	ICU	ASRATT debug	1773	06ED	ADMKTSWI	IMS	Called routine address 0
1742	06CE	ADMPWCMD	ICU	ASRFMT debug	1780	06F4	ADMNUMER	NMR	Invalid parameter
1743	06CF	ADMPDECO	ICU	invalid operation	1781	06F5	ADMYERRM	CSI	Invalid call arguments
1744	06D0	ADMPLSRT	ICU	invalid row or col	1782	06F6	ADMYERRM	CSI	Invalid key
1745	06D1	ADMPRECT	ICU	invalid operation	1783	06F7	ADMYERRM	CSI	Duplicate use of retrieve
1746	06D2	ADMPIGIO	ICU	invalid graphics input data	1784	06F8	ADMYERRM	CSI	Duplicate use of release
1747	06D3	ADMPIGUN	ICU	invalid chart note function	1785	06F9	ADMYERRM	CSI	Invalid call arguments
1748	06D4	ADMPWNOT	ICU	expected column heading field on chart notes panel does not exist	1786	06FA	ADMYERRM	CSI	Invalid key
1749	06D5	ADMPWNOT	ICU	column content too long for this module or field on screen	1790	06FE	ADM3ARO	IMC	Unsupported RCP code
1750	06D6	ADMOQPU	OPU	Invalid request code	1791	06FF	ADM3ARO ADM3IRO ADM3PRO ADM3PROJ ADM3RO	IMC	Invalid RCP code (component or sub component)
1751	06D7	ADMOQPU	OPU	Error issuing DSQDEV	1792	0700	ADM3ARO ADM3RO	IMC	Invalid control block

Dec	Hex	Issuing module	Comp	Description	Dec	Hex	Issuing module	Comp	Description
1793	0701	ADM3AASL ADM3AMCP ADM3AMCV ADM3AMEX ADM3AMRH ADM3AMRV ADM3AMSC ADM3AQQL ADM3AQQR ADM3AQQS ADM3AQQW ADM3AQRQ ADM3ATGI ADM3ATGS ADM3ATGT ADM3ATIM ADM3ATPI ADM3ATPS ADM3ATPT ADM3ATRF	IMC	IMC NIF logic error	1908	0774	AEMIOS00	IMD	Sequential request type not recognized
		ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI			1910	0776	AEMIOS01	IMD	Open request not recognized
		ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI			1915	077B	AEMSRV02	IMD	Function requested not recognized
		ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI			1940	0794	ADMJGSM	GKS	Invalid segment update
		ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI			1950	079E	ADMDMOMD	FSM	Module called with invalid parameters
		ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI			1952	07A0	ADMDMOQ	FSM	Module called with invalid parameters
1795	0703	ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI	IMC	Illegal call from presentation service	1953	07A1	ADMDMOR	FSM	Module called with invalid parameters
		ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI			1955	07A3	ADMDMOPD	FSM	Module called with invalid parameters
		ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI			1956	07A4	ADMDMDMD ADMDMDMD	FSM	Module called with invalid parameters
		ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI			1958	07A6	ADMDMOC	FSM	Module called with invalid parameters
		ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI			1959	07A7	ADMDMSLG	FSM	Module called with invalid parameters
1800	0708	ADMIHELP ADMISSCH ADMISSPR ADMISTEP ADMISYMR	ISE	Invalid interrupt type from ASREAD	1960	07A8	ADMDMODG	FSM	Module called with invalid parameters
1801	0709	ADMISSCH ADMISTEP ADMISYMR	ISE	Unknown alphanumeric field number retrieved by ASQMOD	1961	07A9	ADMDMR	FSM	Module called with invalid parameters
1802	070A	ADMIFSM	ISE	Unknown component type for ADMIFSM	1962	07AA	ADMDMCT	FSM	Module called with invalid parameters
1803	070B	ADMISYME	ISE	Invalid component code	1970	07B2	ADMDMI	FSM	Module called at wrong time
1870	074E	ADMGSTBI	IMS	Invalid utility ID	1971	07B3	ADMDMOMD	FSM	Module called at wrong time
1901	076D	AEMIOS03	IMD	Receive request not recognized	1972	07B4	ADMDMODG	FSM	Module called at wrong time
1902	076E	AEMIOS03	IMD	Screen send request not recognized	1974	07B6	ADMDMOGO	FSM	Module called at wrong time
1903	076F	AEMIOS03	IMD	Send request not recognized	1991	07C7	ADMDMSMV ADMDMSMV	FSM	Output buffer too small
1904	0770	AEMIOS03	IMD	Request was not send or receive	1992	07C8	ADMDMOFM	FSM	Unable to find Symbol Set Table entry after load

abends

Appendix C. Message-to-module cross-reference

This appendix lists the messages issued by GDDM Base, GDDM-PGF, GDDM-IVU, GDDM-GKS, and GDDM-IMD in numeric order with the modules that issue the messages. The full text of the messages and an explanation of each is given in the *GDDM Messages* book. The GDDM-IMD messages start on page 195.

GDDM Base, GDDM-PGF, GDDM-IVU, and GDDM-GKS messages start with the letters ADM, followed by four digits. GDDM-IMD messages start with the letters AEM, followed by five digits. The three letters with which each of these messages starts are omitted from this list to aid clarity.

GDDM-REXX, GDDM-PCLK, and GDDM-OS/2 Link messages are not shown in the cross-reference list.

Msg	Modules that issue the message				Msg	Modules that issue the message			
0001	ADMAC0	ADMASEPB	ADMASP	ADME000C	0084	ADMDSDS			
	ADME000I				0085	ADMLIN1C	ADMLIN1I	ADMLIN1O	ADMLIN1T
0002	ADME000I					ADMLIN1V	ADMLIN2C	ADMLIN2I	ADMLIN2T
0003	ADMASEPB	ADME000C	ADME000I			ADMLIN2V	ADMLIN3C	ADMLIN3I	ADMLIN3T
0004	ADMASP	ADME000C	ADME000I			ADMLIN3V	ADMLIN4D	ADMLIN4T	ADMLIN4V
0005	ADMASP	ADME000C	ADME000I		0086	ADMDSF1	ADMDSF1	ADMLIN1T	ADMLIN1V
0006	ADMASP	ADME000C	ADME000I			ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0007	ADME000C				0087	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0008	ADME000C					ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0009	ADMAEMP					ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0010	ADMAEMP				0088	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0014	ADMAC0				0089	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0040	ADMACFP					ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0041	ADMACFP				0090	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0042	ADMADQP				0091	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0043	ADMACPG	ADMUOTT	ADMUOTV		0092	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0044	ADMADQP				0093	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0048	ADMACUP				0094	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0049	ADMACUP				0095	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0050	ADMACPV	ADMACUP	ADMACUPS	ADMATPE	0096	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0051	ADMAC0	ADMUOTT	ADMUOTV		0097	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0052	ADMAC0				0098	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0053	ADMACUPS				0099	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0054	ADMAC0	ADMUOTT	ADMUOTV		0100	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0056	ADMAC0				0101	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0057	ADMACUP				0102	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0058	ADMACUP				0103	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0059	ADMACUP				0104	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0060	ADMACUP				0105	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0061	ADMACUP	ADMACUPS			0106	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0062	ADMACUP	ADMACUPS			0107	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0063	ADMACUPS				0108	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0064	ADMACUP				0109	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0065	ADMACUP				0110	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0066	ADMACUP	ADMACUPS			0111	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0067	ADMAC0				0112	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0068	ADMACFP				0113	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0069	ADMACUPS				0114	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0070	ADMDSRO				0115	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0071	ADMDSDO				0116	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0072	ADMDSDO				0117	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0073	ADMDSDO				0118	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0074	ADMDSDS				0119	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0075	ADMDSDS				0120	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0076	ADMDSDS				0121	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0077	ADMDSDS				0122	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0078	ADMDSDS				0123	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0079	ADMDSDS	ADMLIN1I	ADMLIN2I	ADMLIN3I	0124	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0080	ADMDSF1				0125	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0081	ADMDSDS				0126	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I
0082	ADMDSDS	ADMDSRO			0127	ADMDSF2	ADMDSF3	ADMDSF4	ADMLIN2I

GDDM messages — 0001 +

Msg	Modules that issue the message				Msg	Modules that issue the message			
0130	ADMDCCPC				0182	ADMDGGNS	ADMDGQTB		
0131	ADMDCCPC				0183	ADMDGPM			
0132	ADMDCCPC	ADMIFSM	ADMVFSM		0184	ADMDGSE			
0133	ADMDCCPC				0185	ADMDGSE			
0134	ADMDCCPC				0186	ADMDGTR			
0135	ADMDCGS				0187	ADMDGPA			
0136	ADMDCQD				0188	ADMDGREG			
0137	ADMDCCPC				0189	ADMDGTR			
0138	ADMDCCPC				0190	ADMDGTR			
0139	ADMDCQD				0191	ADMDGPI			
0140	ADMDGIN	ADMDGSC	ADMBCHSG	ADMBNOTE	0192	ADMDGPI			
	ADMDGMSV	ADMDGSE	ADMDGSV	ADMDGTR	0193	ADMDGPI			
0141	ADMDGWI				0194	ADMDGTR			
0142	ADMDGWI				0195	ADMDGXCR			
0143	ADMBCHSG	ADMDGMSV	ADMDGSE	ADMDGSV	0196	ADMDGXCR			
0144	ADMDGWI				0197	ADMUPCT	ADMUPCV	ADMUPGT	ADMUPGV
0145	ADMDGPO	ADMDGSC	ADMDGSE	ADMDGTR	0198	ADMDGPA	ADMDGPC	ADMDGPE	ADMDGPPEE
0146	ADMDGPE	ADMDGPPEE	ADMDGLD	ADMDGMLD		ADMDGPM	ADMDGPR		
	ADMDGMSV	ADMDGPO	ADMDGSV	ADMDGGI	0199	ADMDGSE			
	ADMDGIN	ADMDGPAD	ADMDGPC	ADMDGPMK	0200	ADMDAPDT			
	ADMDGPR	ADMDGTR	ADMDGXCR		0201	ADMDAPFA	ADMDAPFC	ADMDAPFD	ADMDAPQC
0147	ADMDGPA					ADMDAPQL			
0148	ADMDGPM				0203	ADMDASFA			
0149	ADMDGSC	ADMDGSE	ADMDGTR		0204	ADMDASFA			
0150	ADMDGSQ	ADMDGPO	ADMDGSC	ADMDGSL	0205	ADMDASCR			
	ADMDGSV	ADMDGLD	ADMDGMLD	ADMDGMSV	0206	ADMDASCR			
	ADMDGPAD	ADMDGSE	ADMDGSV	ADMDGWI	0207	ADMDAPFM	ADMDAPQF		
0151	ADMBCHVU	ADMDGWI			0208	ADMDAPDF	ADMDAPFM	ADMDAPQF	
0152	ADMDGPA	ADMDGPAC			0209	ADMDAPCU	ADMDAPFM	ADMDAPMF	ADMDAPQF
0153	ADMDAPRS	ADMDGSQ	ADMDGEN	ADMDGPI	0211	ADMDAPMF	ADMDASCA		
	ADMDGMSV	ADMDGPAD	ADMDGPO	ADMDGPR	0212	ADMDAPCU	ADMDAPFA	ADMDAPFC	ADMDAPFM
	ADMDGSV	ADMDGWI				ADMDAPMF	ADMDAPQC	ADMDAPQL	
0154	ADMDGPC	ADMDGEN	ADMDGFL	ADMDGFP	0213	ADMDAPQF			
	ADMDGGDD	ADMDGIN	ADMDGLD	ADMDGPA	0214	ADMDAPFC	ADMDAPQC		
	ADMDGPAC	ADMDGPAX	ADMDGPC	ADMDGPE	0215	ADMDAPFC			
	ADMDGPPEE	ADMDGPI	ADMDGPM	ADMDGPMK	0216	ADMDATYP			
	ADMDGPO	ADMDGPR	ADMDGOI	ADMDGRQ	0217	ADMDAPFC	ADMDAPQC		
	ADMDGSE	ADMDGSQ	ADMDGTR	ADMDGWI	0218	ADMDAPER	ADMDAPQF	ADMDAPRS	ADMDAQCU
	ADMDGXC	ADMDGXCR	ADMDGXE	ADMDOGR	0219	ADMDAPFC	ADMPMENU	ADMPWTXT	ADMPWXY
	ADMDWADB	ADMDWPIP	ADMDGPA	ADMDGTR	0220	ADMDAPQM			
0155	ADMDGPA	ADMDGPAC			0221	ADMDAPCU			
0156	ADMBNOTE	ADMDGL4	ADMDGPD6	ADMDGPMK	0222	ADMDAPRM	ADMDQDEF	ADMDQMOD	
	ADMDGPR	ADMDGPT6	ADMDGP1	ADMDGP3	0223	ADMDAPFC			
	ADMDGP4	ADMDGT4			0224	ADMDASFA			
0157	ADMDGPA	ADMDGPAC			0225	ADMDAPFM			
0158	ADMDGPA	ADMDGPAC	ADMDGPOP	ADMDGPC	0226	ADMDASFA			
	ADMDGPM	ADMDGPMK	ADMDGPR		0227	ADMDASFA			
0159	ADMDGPR				0228	ADMDAPFC	ADMPMENU	ADMPWTXT	ADMPWXY
0160	ADMDGPR				0229	ADMDAPFC	ADMPMENU	ADMPWTXT	ADMPWXY
0161	ADMDGGDD	ADMDGPO	ADMDGSC	ADMDGMSV	0230	ADMACPT			
	ADMDGSE	ADMDGSPP	ADMDGSV		0231	ADMACPT			
0162	ADMDGWI				0232	ADMACPT			
0163	ADMDGWI				0233	ADMDGPCX	ADMDHPC	ADMDJCS	ADMUPCT
0164	ADMDGWI					ADMUPCV	ADMUPGT	ADMUPGV	
0165	ADMDGWI				0234	ADMDHPC			
0166	ADMDGWI				0235	ADMACPT			
0167	ADMDGSE				0236	ADMACPT			
0168	ADMDGPA	ADMDGPAC			0244	ADMDIAL	ADMDICO	ADMDIGR	ADMDIIM
0169	ADMDGPC					ADMDITX	ADMDIMP	ADMDIOC	ADMDIPA
0170	ADMDGPM					ADMDISS			
0171	ADMDGPM				0248	ADMDIOC	ADMEGF40		
0172	ADMDGPI	ADMDGPM			0249	ADMDICO			
0173	ADMDGPI	ADMDGPO			0250	ADMDIAL			
0174	ADMDGPI	ADMUPCT	ADMUPCV	ADMUPGT	0257	ADMDKDI	ADMDPUAO		
	ADMUPGV				0259	ADMDKAP	ADMDPUAO		
0175	ADMDGPI	ADMDGPM			0260	ADMDICO	ADMDIPA		
0176	ADMDGPPEE				0261	ADMDIPA			
0177	ADMBPIE	ADMDGPE	ADMDGPPEE		0264	ADMDISS			
0178	ADMDGPO				0270	ADMDSSRD			
0179	ADMDGPO	ADMDGSQ	ADMDGWI	ADMDGGP	0272	ADMDDPWR	ADMDKSH		
	ADMDGLD	ADMDGMLD	ADMDGMSV	ADMDGPAD	0273	ADMDPC	ADMDPSCD	ADMDWC	
	ADMDGSE	ADMDGSV	ADMDGTR	ADMDOCO	0274	ADMDDIMF	ADMDDIMG	ADMDDIMX	ADMDUBC
	ADMDGSC					ADMDKTX	ADMDSEH	ADMDXSCD	
0180	ADMDGMSV	ADMDGPO	ADMDGSL	ADMDGSV	0275	ADMDLC	ADMDPC	ADMDWC	ADMD3DIF
0181	ADMDGPI	ADMDGPM				ADMD3EMU	ADMD3SDS	ADMD3SXF	

Msg	Modules that issue the message			
0276	ADMDJC	ADMDKC	ADMDLC	ADMDOCO
	ADMDPC	ADMDTC	ADMDWC	ADMD3SDE
0277	ADMDJC	ADMDKC	ADMDLC	ADMDOCO
	ADMDPC	ADMDSLD	ADMDTC	ADMDWC
0279	ADMDWC	AEMIOS03		
0281	ADMDOOC	ADMDSPQ		
0282	ADMDICO	ADMDOLO		
0283	ADMDOCO			
0284	ADMDOCO			
0285	ADMDOIM			
0286	ADMDJC	ADMDLC	ADMDOCO	ADMDRC
	ADMDWC			
0287	ADMDJC	ADMDLC	ADMDOCO	ADMDRC
	ADMDWC	ADMDJXMJ		
0300	ADMKTSWI	ADMDJCX	ADME000C	ADME000I
	ADM3IASV	ADM3IGT	ADMACSSP	ADMECM0V
	ADMEGFLO	ADMEGFOD	ADMEROU	ADMER00C
	ADMER00D	ADMER00O	ADMER00V	ADMIESI
	ADMJGSM	ADMLIN1I	ADMLIN2I	ADMLIN3I
	ADMLRN1I	ADMPIDIF	ADMYERRM	ADM3AAAR
	ADM3AMRM	ADM3ATBF	ADM3ATPT	ADM3ATTFF
	ADM5ERR	ADM5INIT		
0301	ADMEOSLC	ADMEOSLD	ADMEOSLO	
0302	ADMEGF0C	ADMEGF0G	ADMEGLGO	ADMEPRGI
	ADMEPR0D	ADMEPR0O	ADMEPR0V	
0303	ADMEGF30	ADMEGF80		
0304	ADMEGFAC	ADMEGFBC	ADMEGF0C	ADMEGFDO
	ADMEGF0V	ADMEGF0C	ADMEGF0G	ADMEGFLO
	ADMEGFOD	ADMEGF0D	ADMEGF0S	ADMEGF12
	ADMEGF14	ADMEGF30	ADMEGF60	ADMEGF80
	ADMEGLEV	ADMEGLFC	ADMEGLGO	ADMEGLII
	ADMEOSD0			
0305	ADMEGF0C	ADMEGF0G	ADMEGLGO	
0306	ADMEGF0C	ADMEGF0D	ADMEGLFC	ADMEOSQ0
	ADMUPCT	ADMUPCV	ADMUPGT	ADMUPGV
0307	ADMACUP	ADMDMSLG	ADMEGFBC	ADMEGFDO
	ADMEGF0V	ADMEGF0C	ADMEGF0G	ADMEGFII
	ADMEGFJI	ADMEGFQD	ADMEGF0S	ADMEGF30
	ADMEGLFC	ADMEOSQ0	ADMIESI	AEMIOS00
	ADMDCOS	ADMIFSM	ADMVFSM	
0309	ADMER00C	ADMER00D	ADMER00O	ADMER00V
0310	ADMEGF0C	ADMEGF0V	ADMEGF0C	ADMEGF0G
	ADMEGFLO	ADMEGF0D	ADMEGFQD	ADMEGLEV
	ADMEGLGO	ADMPCKEK	ADMEGF0V	ADM1MSLV
	ADMEPQ00			
0311	ADMEGF0D	ADMEGF0D	ADMER00C	ADMER00D
	ADMER00O	ADMER00V		
0312	ADMEGFAC	ADMEGFBC	ADMEGF0C	ADMEGF0V
	ADMEGF0C	ADMEGF0G	ADMEGFII	ADMEGFLO
	ADMEGFOD	ADMEGF0D		
0313	ADMDCFW	ADMDCGS	ADMDMSLG	ADMDSCC
	ADMEGF0D	ADMEGF30	ADMEGF40	ADMEGF70
	ADMEGF80	ADMEOSD0	ADMPGET	ADMP0BJS
	ADM3IASV	ADM3PRST		
0314	ADMACUP	ADMEGF0C	ADMEGF0D	
0315	ADMEGF0V	ADMLTM1V		
0316	ADMEGF0V			
0317	ADMEGFAC	ADMEGF0C	ADMLTM30	ADMLTM40
0318	ADMEGF0C	ADMEGF0G	ADMEGLGO	
0319	ADMEGF10	ADMEPR0C		
0320	ADMEGF0C	ADMEGF0G	ADMEGFLO	ADMEGF0D
	ADMEGLGO	ADMEPQ00		
0321	ADMEPQ00			
0322	ADMEGF0V	ADMEGF70	ADMEGLFC	
0323	ADMEGF0V	ADMEGF0C	ADMEGFLO	ADMEGF0D
	ADMEGFQD			
0324	ADMDGMSV	ADMEGFBC	ADMEGF0D	ADMEGF0V
	ADMEGF0C	ADMEGF0G	ADMEGFII	ADMEGF0D
	ADMEGFQD	ADMPSAVE	AEMIOS00	
0325	ADMACPG	ADMACUP	ADMDCFW	ADMDPPWR
	ADMDGLD	ADMDJUT	ADMDKSH	ADMDMSLG
	ADMDSCC	ADMDSPDD	ADMDSSPF	ADMEGFBC
	ADMEGF0C	ADMEGF0V	ADMEGF0C	ADMEGF0G
	ADMEGFII	ADMEGFJI	ADMEGFLO	ADMEGF0D

Msg	Modules that issue the message			
	ADMEGF0D	ADMEGFQD	ADMEGF30	ADMEGF60
	ADMEGF80	ADMEGLEV	ADMEGLFC	ADMEGLGO
	ADMEGLII	ADMEOSD0	ADMF0U	ADMLRC20
	ADMP0RT	ADMPGET	ADMPIMP	ADMPIS0Q
	ADMP0BJS	ADMUPCT	ADMUPCV	ADMUPGT
	ADMUPGV	ADM3IASV	ADM3PRST	ADM4GSF
	ADM5IM	AEMIOS03	ADMEGF0V	ADMEGFJI
0326	ADMEDCKI	ADMEDK0C	ADMEDK0O	ADMEDK0V
0327	ADMEGFAC	ADMEGFBC	ADMEGF0C	ADMEGF10
	ADMEGF13	ADMEGLFC	ADMLRN2C	ADMLTM2C
0328	ADMEGF0D	ADMEGF0S	ADM5IM	
0329	ADMEPQ00			
0330	ADMEPQ00			
0334	ADMEPCBI	ADMLIN2I	ADMLIN3I	
0335	ADMEDPCI			
0336	ADMEDPCI			
0338	ADMEDLII	ADMEGLII		
0339	ADMEGFJI			
0340	ADMEGFJI			
0341	ADMESDSI			
0342	ADMESDSI			
0343	ADMESDSI			
0344	ADMEGFII	ADMEOSLI		
0347	ADMEDLII			
0348	ADMEGFII	ADMEGFJI		
0350	ADMEGF0G			
0360	ADMEGF0G	ADMEGFLO	ADMEGF0D	
0361	ADMEGF10			
0362	ADMEGF0G			
0370	ADMACPG	ADMEOSLC	ADMEOSLD	ADMEOSLI
	ADMEOSLO	ADMEOSQ0	ADMU0TT	ADMU0TV
	ADMEOSLC	ADMEOSLD	ADMEOSLI	ADMEOSLO
0371	ADMEGF60			
0372	ADMEPRGI	ADMEPR0C	ADMEPR0D	ADMEPR0O
0373	ADMEPRGI	ADMEPR0C	ADMEPR0D	ADMEPR0O
0374	ADMEGF60			
0400	ADMLIN2I	ADMLQU1C	ADMLQU1I	ADMLQU1O
	ADMLQU1T	ADMLQU1V		
0401	ADMLIN1T	ADMLTM1T		
0402	ADMLQU1T	ADMLSN1T		
0403	ADMLIN1T	ADMLQU1T	ADMLRC1T	ADMLSN1T
	ADMLTM1T			
0404	ADMDPPWR	ADMD0UBC	ADMDKRX	ADMDSF1
	ADMDSPDD	ADMDSQA	ADMDSQL	ADMDSQP
	ADMDSXP	ADMDXRIO	ADMLBC1V	ADMLQU1C
	ADMLQU1O	ADMLQU1T	ADMLQU1V	
0405	ADMLRC1T	ADMLSN1T		
0406	ADMDTC	ADMDTCON	ADMDTPGD	ADMLFM10
	ADMLRC1I	ADMLRC1V	ADMLRO00	
0407	ADMLQU1C	ADMLQU1I	ADMLQU1O	ADMLQU1T
	ADMLQU1V			
0408	ADMLAC1I	ADMLRN2C	ADMLRN2I	ADMLRN2V
	ADMLSN20			
0409	ADMLIN4D	ADMLIN4T	ADMLIN4V	
0410	ADMLER1O	ADMLIN1O	ADMLQU1O	ADMLSN1O
	ADMLXR1O			
0411	ADMLER1O			
0412	ADMLER1O			
0413	ADMLER1O			
0414	ADMLER1O			
0415	ADMLER1O			
0416	ADMLER1O			
0417	ADMLER1O			
0418	ADMLER1O			
0419	ADMLER1O			
0420	ADMLER1O			
0421	ADMLER1O			
0422	ADMLER1O	ADMLIN1O		
0423	ADMLER1O			
0424	ADMLER1O			
0425	ADMLER1O	ADMOQPU		
0426	ADMLER1O			
0427	ADMLER1O			
0428	ADMLXR1C	ADMLXR1O		
0435	ADMLIN1T			
0436	ADMLIN1T			

Msg	Modules that issue the message			
0437	ADMLSN1T			
0438	ADMLIN2T			
0440	ADMLRC1V			
0441	ADMLER1V			
0443	ADMLER1V	ADMLQU1V		
0444	ADMLIN1V			
0445	ADMLIN1V			
0446	ADMLRN2V	ADMLTM2V		
0447	ADMLIN1V	ADMLN4V		
0448	ADMLIN1V	ADMLN4V		
0449	ADMLIN1V	ADMLN4V		
0460	ADMLRO1I	ADMLRO2I	ADMLRO3I	
0461	ADMLIN1I			
0462	ADMLIN1I	ADMLIN2I	ADMLIN3I	
0463	ADMLSN1I			
0470	ADMLRN3C			
0481	ADMLIN1C	ADMLIN1I	ADMLIN1O	ADMLIN1T
	ADMLIN1V	ADMLIN2C	ADMLIN2I	ADMLIN2T
	ADMLIN2V	ADMLIN3C	ADMLIN3I	ADMLIN3T
	ADMLIN3V	ADMLIN4D	ADMLIN4T	ADMLIN4V
0482	ADMLIN1C	ADMLIN1T	ADMLIN1V	ADMLIN2C
	ADMLIN2I	ADMLIN2T	ADMLIN2V	ADMLIN3I
	ADMLIN3V	ADMLIN4D	ADMLIN4T	ADMLIN4V
0483	ADMLQU40			
0484	ADMLIN1C	ADMLIN1I	ADMLIN1T	ADMLIN1V
0485	ADMLIN1C	ADMLIN1T	ADMLIN1V	ADMPINIT
0486	ADMLIN1C	ADMLIN1O		
0489	ADMLIN4T	ADMLIN4V		
0490	ADMLER1O	ADMLQU1O	ADMLSN1O	ADMLXR1O
0491	ADMLIN1O	ADMLQU1O	ADMLSN1O	ADMLXR1O
	ADMOPST			
0495	ADMLXR1C	ADMLXR1O	ADMOQPU	
0496	ADMDDPWR	ADMDDUBC	ADMCKTX	ADMDSQA
	ADMDSQL	ADMDSQP	ADMDSXP	ADMWTRM
	ADMDEXR1O	ADMLCN1C	ADMLER1O	ADMLQU1C
	ADMLQU1O	ADMLQU1T	ADMLQU1V	ADMLRC1T
	ADMLSN1C	ADMLSN1O	ADMLSN1T	ADMLXR1C
	ADMLXR1O			
0497	ADMCKTX	ADMLBC1V	ADMLXR1C	
0498	ADMCKRX	ADMCKUT		
0499	ADMDDPWR	ADMDDUBC	ADMCKTX	ADMDSQA
	ADMDSQL	ADMDSQP	ADMDSXP	ADMWTRM
	ADMDEXR1O	ADMLBC1V	ADMLCN1C	ADMLQU1C
	ADMLQU1T	ADMLQU1V	ADMLRC1T	ADMLSN1C
	ADMLSN1O	ADMLSN1T		
0501	ADMBBGS	ADMBCHRT	ADMBGCHR	ADMBGFIX
	ADMBGFLT			
0502	ADMBDRAW			
0503	ADMBBGS	ADMBGCHR	ADMBGFIX	ADMBGFLT
0504	ADMBDRAW			
0505	ADMBSET			
0506	ADMBASEL			
0507	ADMBSET			
0508	ADMBGCHR	ADMBGFIX	ADMBGFLT	ADMBMPRJ
	ADMBSET			
0509	ADMBGCHR	ADMBNOTE	ADMBSET	
0510	ADMBDKEY			
0511	ADMBDKEY			
0512	ADMBDRAW			
0513	ADMBSET			
0514	ADMBLABL	ADMBNOTE		
0515	ADMBSET			
0516	ADMBGCHR	ADMBGFIX	ADMBGFLT	ADMBMPRJ
	ADMBSET			
0517	ADMBSET			
0518	ADMBSET			
0519	ADMBSET			
0520	ADMBSET			
0521	ADMBSET			
0522	ADMBSET			
0523	ADMBCHCV	ADMBGFIX	ADMBGFLT	ADMBMPRJ
	ADMBQPOS	ADMBSET		
0524	ADMBGFIX	ADMBGFLT	ADMBMPRJ	ADMBSET
0525	ADMBCHSG			

Msg	Modules that issue the message			
0526	ADMBDRAX			
0527	ADMBDRAX			
0528	ADMBDRAX			
0529	ADMBARS			
0530	ADMBADTM	ADMBDSAX		
0531	ADMBADTM	ADMBDOAX	ADMBNOTE	
0532	ADMBCHRT			
0533	ADMBCHRT			
0534	ADMBARS	ADMBDTAB	ADMBHIST	ADMBMANH
	ADMBPLOT	ADMBPOLR		
0535	ADMBARS	ADMBCHRT	ADMBDTAB	ADMBHIST
	ADMBPLOT			
0536	ADMBARS			
0537	ADMBPIE			
0538	ADMBPIE			
0539	ADMBPIE			
0540	ADMBPIE			
0541	ADMBVENN			
0542	ADMBVENN			
0543	ADMBCHRT			
0544	ADMBCHRT	ADMBGFTX	ADMBMISS	ADMBPOLR
0545	ADMBDRAX			
0546	ADMBCHVU	ADMBDRAX	ADMBDTAB	
0547	ADMBARS			
0548	ADMBCHSG	ADMBNOTE		
0549	ADMBSET			
0550	ADMBARS	ADMBCHRT	ADMBDKEY	ADMBDTAB
	ADMBMISS	ADMBSET		
0551	ADMBNOTE			
0552	ADMBNOTE			
0553	ADMBNOTE			
0554	ADMBSET			
0555	ADMBSET			
0556	ADMBNOTE			
0557	ADMBDTTL			
0558	ADMBDTTL			
0559	ADMBVENN			
0560	ADMBDRAX			
0561	ADMBPIE			
0562	ADMBPIE	ADMBVENN		
0563	ADMBNOTE			
0564	ADMBNOTE			
0565	ADMBNOTE			
0566	ADMBDSAX			
0567	ADMBNOTE			
0568	ADMBQPOS			
0569	ADMBSET			
0570	ADMBMPRJ			
0571	ADMBPIE			
0572	ADMBPIE			
0573	ADMBMANH			
0574	ADMBMANH			
0575	ADMBQPOS			
0576	ADMBCHCV			
0577	ADMBCHCV			
0578	ADMBCHCV			
0579	ADMBCHCV	ADMBNOTE		
0580	ADMBCHCV			
0581	ADMBMANH			
0582	ADMBMANH			
0583	ADMBDOAX	ADMBDSO		
0584	ADMBDRAX			
0585	ADMBDRAW			
0586	ADMBNOTE			
0587	ADMBDTAB			
0588	ADMBDTAB			
0589	ADMBLABL			
0601	ADMVSELP	ADMVSymb		
0603	ADMVASPE	ADMVCC	ADMVERAS	ADMVREF
	ADMVREN	ADMVSELP	ADMVSWIT	
0604	ADMVSELP			
0605	ADMVSELP			
0606	ADMVVSSE			
0609	ADMVSELP			
0610	ADMVSELP			

Msg	Modules that issue the message			
0611	ADMVBREA	ADMVJOIN		
0612	ADMVBREA	ADMVJOIN		
0615	ADMVREN			
0616	ADMVREN			
0617	ADMVREN			
0618	ADMVSYMB			
0619	ADMVSYMB			
0620	ADMVSYMB			
0621	ADMVSYNT			
0622	ADMVVSSE			
0623	ADMVSYNT			
0624	ADMVSAVE			
0625	ADMVSYNT			
0626	ADMVSYNT			
0627	ADMVSYNT			
0628	ADMVSYNT			
0629	ADMVJOIN			
0630	ADMVSYNT			
0632	ADMVSYNT			
0633	ADMVSYNT			
0634	ADMVSYMB	ADMVSYNT		
0635	ADMVSYNT			
0636	ADMVSYNT			
0637	ADMVSYNT			
0638	ADMVSGET			
0639	ADMVSAVE			
0640	ADMVFIN			
0642	ADMVSELP			
0645	ADMVSHAD			
0646	ADMVSSSEC	ADMVSSSET	ADMVSSSEV	
0647	ADMVSSSE			
0649	ADMVCC			
0651	ADMVBACK	ADMVCHAN	ADMVCOPY	ADMVCOP1
	ADMVDEL	ADMVSHIF	ADMVSTRE	
0652	ADMVBOUN			
0653	ADMVBOUN	ADMVGRIN	ADMVSHIF	
0654	ADMVUPDC			
0657	ADMVCC			
0660	ADMVCC			
0661	ADMVEXIT			
0671	ADMVHELP			
0672	ADMVHELP			
0673	ADMVHELP			
0674	ADMVHELP			
0675	ADMVHELP			
0676	ADMVHELP			
0677	ADMVGRIN	ADMVHELP	ADMVSELP	ADMVSYMB
0678	ADMVHELP			
0679	ADMVHELP			
0680	ADMVHELP			
0685	ADMVCC	ADMVSAVE	ADMVSETR	
0691	ADMVSAVE			
0697	ADMVSETR			
0700	ADMPINIT			
0701	ADMPDRAP			
0702	ADMPINIT			
0703	ADMPGXY			
0704	ADMPINIT			
0705	ADMPGXY			
0706	ADMPGXY	ADMPINIT		
0707	ADMPINIT			
0708	ADMPGXY	ADMPRXY		
0709	ADMPGXY			
0711	ADMPGET			
0712	ADMPGET			
0713	ADMPSAVE			
0714	ADMPDRAM	ADMPDRAP	ADMPDRAW	ADMPSAVE
0716	ADMPINIT			
0717	ADMPINIT			
0718	ADMPGXY			
0719	ADMPINIT			
0720	ADMPMAIN			
0721	ADMPMAIN			
0722	ADMPMAIN			
0723	ADMPMAIN			

Msg	Modules that issue the message			
0724	ADMPMAIN			
0725	ADMPBCHC	ADMPBCSQ	ADMPBDIR	ADMPBIFC
	ADMPBNUM	ADMPMAIN		
0726	ADMPBCSQ	ADMPMAIN		
0727	ADMPBCSQ	ADMPBDAT	ADMPBDIR	ADMPMAIN
0728	ADMPBIFC			
0729	ADMPBCHC	ADMPBIFC		
0730	ADMPBIFC			
0731	ADMPBIFC			
0732	ADMPBCHC			
0733	ADMPBDIR			
0734	ADMPBDAT			
0735	ADMPBDAT			
0736	ADMPBCHC	ADMPBIFC	ADMPBNUM	
0737	ADMPBCSQ			
0738	ADMPBNUM			
0739	ADMPBDAT			
0740	ADMPBCHC	ADMPBIFC		
0741	ADMPBCHC	ADMPBIFC		
0742	ADMPBDAT			
0743	ADMPBCHC	ADMPBIFC		
0751	ADMOQPU			
0754	ADMOQPU			
0758	ADMDICO			
0760	ADMLQU10	ADMYSSSTB		
0761	ADMYRSRL			
0762	ADMYRSRL			
0767	ADMYDTIM			
0770	ADMKSCHD			
0771	ADMKNEWM			
0772	ADMKNEWM			
0773	ADMKNEWM			
0774	ADMKOLDM			
0775	ADMKSRVC			
0776	ADMKNEWM			
0801	ADMISHFT	ADMISTEP		
0802	ADMISSCH	ADMISSPR	ADMISTEP	ADMISYMR
0804	ADMISYNT			
0805	ADMISSCH	ADMISTEP	ADMISYMR	ADMISYNT
0807	ADMIHELP	ADMISSCH	ADMISSPR	ADMISTEP
	ADMISYMR			
0808	ADMISTEP			
0809	ADMISTEP			
0810	ADMISTEP	ADMISYNT		
0811	ADMISSCH			
0812	ADMIBNSS			
0813	ADMISSCH			
0814	ADMISSCH			
0815	ADMISSCH			
0816	ADMISSCH			
0817	ADMISAVE			
0818	ADMIBNSS			
0819	ADMIBNSS			
0820	ADMISYNT			
0821	ADMISAVE			
0822	ADMIDMCH	ADMISHFT		
0823	ADMISSPB	ADMISSPR		
0824	ADMISSPB	ADMISSPR		
0825	ADMISYMB	ADMISYMR		
0827	ADMIBHED			
0828	ADMISYNT			
0829	ADMISYNT			
0830	ADMISYNT			
0831	ADMICOMP	ADMISYNT		
0832	ADMIGLOB	ADMISYMR	ADMISYNT	
0833	ADMIBNSS			
0834	ADMIBNSS			
0835	ADMIHELP	ADMISSCH	ADMISSPR	ADMISTEP
	ADMISYMR			
0836	ADMIBNSS			
0837	ADMISSCH			
0838	ADMISSPR	ADMISYMR		
0839	ADMIREF			
0840	ADMISSCH	ADMISYMR		
0842	ADMICURS			

GDDM messages — 0001 +

Msg	Modules that issue the message					Msg	Modules that issue the message				
0843	ADMICURS					1023	ADMPRECT				
0844	ADMICHNG	ADMISYMR				1024	ADMPLFIT	ADMPRCMD	ADMPTXTL		
0845	ADMISYMR					1025	ADMPDRAX				
0847	ADMISTEP					1026	ADMPRCMD				
0848	ADMIBHED					1027	ADMPRCMD				
0849	ADMIBHED					1028	ADMPRCMD				
0851	ADMIBNSS					1029	ADMPRCMD				
0852	ADMIFILL					1030	ADMPCKEY				
0853	ADMIFILL					1032	ADMPRCMD				
0854	ADMIFILL	ADMISYMR				1033	ADMPRCMD				
0855	ADMISYMR					1034	ADMPRCMD				
0857	ADMIFILL	ADMISYMR				1035	ADMPRCMD				
0858	ADMIFILL	ADMISYMR				1036	ADMPRCMD				
0859	ADMIFILL	ADMISYMR				1037	ADMPRCMD				
0860	ADMISYME					1038	ADMPRCMD				
0861	ADMISYMR					1039	ADMPRCMD				
0862	ADMISTEP					1044	ADMPDRIR				
0872	ADMDSF1					1045	ADMPICDR	ADMPRCMD			
0875	ADMDSPPD	ADMDSQP	ADMDSXP	ADMDCSV		1046	ADMPRCMD				
0876	ADMDDPWR	ADMDDUBC	ADMDSPPD	ADMDSQP		1047	ADMPRATT				
0877	ADMDDUBC	ADMDTCON	ADMDTCON	ADMDTPGD		1053	ADMPRINT				
0878	ADMDDUBC	ADMDSQP				1054	ADMPRINT				
0879	ADMDSXP					1055	ADMPRINT				
0880	ADMDSKQ					1056	ADMPRINT				
0881	ADMDSF1					1057	ADMPRINT				
0882	ADMDSF4					1058	ADMPRINT				
0889	ADMDOCO					1061	ADMPCKEY				
0890	ADMDOCO					1062	ADMPDRAS				
0891	ADMDOCO					1063	ADMPDRAS				
0892	ADMDOCO					1064	ADMPRMLT				
0893	ADMDOCO					1065	ADMPICABS	ADMPCEXC	ADMPICDR	ADMPICSEL	
0894	ADMDOCO						ADMPICSR	ADMPICSUM	ADMPICCMD		
0895	ADMDOCO					1066	ADMPINDX				
0896	ADMDOCO					1067	ADMPINDX				
0897	ADMDOCO					1068	ADMPINDX				
0898	ADMDOCO	ADMDCOU				1069	ADMPRATT				
0899	ADMDOCO	ADMDCOU				1070	ADMPRATT				
0900	ADMDCIN					1071	ADMPLFIT				
0901	ADMDCIN					1072	ADMPLFIT				
0904	ADMDCDS	ADMDCPS	ADMDCQS	ADMDCDOC		1073	ADMPLCMD				
	ADMDOCO	ADMDSLD				1074	ADMPRCMD				
0906	ADMDCS4	ADMDCJUT				1075	ADMPLCMD	ADMPLFIT			
0911	ADMDCJCH					1076	ADMPLCMD				
0912	ADMDCGS					1077	ADMPINDX				
0913	ADMDCJDOC					1078	ADMPINDX				
0914	ADMDCJDOC					1079	ADMPRINT				
0915	ADMDCJCX					1080	ADMPRINT				
0920	ADMDC	ADMDCCON	ADMDCPGD	ADMOQPU		1081	ADMPRINT				
	ADMPRINT					1082	ADMPICUN				
0921	ADMDC	ADMDCCON	ADMDCPGD	ADMPRINT		1083	ADMPICUN				
0922	ADMDC	ADMDCCON	ADMDCPGD			1084	ADMPICUN				
0923	ADMDC	ADMDCCON	ADMDCPGD	ADMPRINT		1085	ADMPICUN				
0925	ADMDC					1086	ADMPICUN				
0968	ADMDCSCR					1087	ADMPICUN				
0999	ADMDCMD	ADMDCRD				1090	ADMPFUNC				
1000	ADMPANEL	ADMPICIO				1091	ADMPDRAM				
1001	ADMPCKEY	ADMPRXY				1092	ADMPICIO				
1002	ADMPCKEY	ADMPRATT	ADMPRMLT	ADMPRNOT		1093	ADMPRATT				
1003	ADMPICDR	ADMPRCMD				1094	ADMPDRAX				
1004	ADMPCKEY					1097	ADMPDRAS				
1005	ADMPCKEY	ADMPRMLT	ADMPRNOT			1098	ADMPMENU	ADMPWTXT	ADMPWXY		
1006	ADMPCKEY					1099	ADMPRCMD				
1007	ADMPCKEY	ADMPRMLT				1100	ADMPCKEY	ADMPFUNC	ADMPGOTO		
1009	ADMPANEL	ADMPICIO				1101	ADMPGOTO				
1010	ADMPRCMD					1102	ADMPGOTO				
1011	ADMPRATT					1103	ADMPGOTO				
1012	ADMPRATT					1104	ADMPGOTO	ADMPRCBE			
1013	ADMPRATT					1105	ADMPRCMD				
1014	ADMPRATT					1106	ADMPFUNC				
1015	ADMPRCMD					1107	ADMPRXY				
1016	ADMPRCMD					1108	ADMPRATT	ADMPWATT			
1017	ADMPDRAM					1109	ADMPCKEY	ADMPGET			
1019	ADMPCKEY					1110	ADMPDRAM				
1020	ADMPCKEY					1118	ADMPICIO				
						1119	ADMPICIO				

Msg	Modules that issue the message			
1128	ADMPIGDA			
1134	ADMPIGDA			
1140	ADMPIGDA			
1143	ADMPIGIO			
1146	ADMPIGDA			
1149	ADMPIMP			
1150	ADMPIDIF	ADMPSEQ		
1154	ADMPIMP			
1155	ADMPIMP			
1156	ADMPRIV			
1157	ADMPRIV			
1158	ADMPRIV			
1163	ADMPIMP	ADMPRIV	ADMPWIV	
1165	ADMPRCMD			
1166	ADMPRIV			
1167	ADMPIMP			
1168	ADMPRIV			
1169	ADMPWIV			
1170	ADMPWIV			
1171	ADMPRIV	ADMPWIV		
1173	ADMPCHK	ADMPSEQ	ADMPWIV	
1174	ADMPFUNC			
1175	ADMPIMP			
1176	ADMPRIV			
1179	ADMPRIV			
1180	ADMPIDIF			
1182	ADMPIMP			
1183	ADMPIMP			
1184	ADMPIMP			
1185	ADMPIMP			
1186	ADMPIMP			
1187	ADMPIMP			
1188	ADMPIMP			
1189	ADMPIMP			
1190	ADMPIMP			
1191	ADMPIMP			
1192	ADMPCHK			
1199	ADMPGET			
1205	ADMPRCBE			
1206	ADMPRCBE			
1209	ADMPRCBE			
1210	ADMPRWSH			
1220	ADMPRWSH			
1221	ADMPDRSS			
1223	ADMPDRSS			
1224	ADMPIGUS			
1225	ADMPIGUS			
1226	ADMPDRSS			
1227	ADMPDRSS			
1228	ADMPDRAM			
1229	ADMPDRAM			
2001	ADMOPUT			
2002	ADMOPUT			
2003	ADMOPUJ	ADMOPUT		
2004	ADMOPUT			
2005	ADMOPUT			
2007	ADMOPUT			
2009	ADMOPUT			
2010	ADMOPUJ	ADMOPUT		
2011	ADMOPUT			
2012	ADMOPUJ	ADMOPUT		
2013	ADMOPUJ	ADMOPUT		
2014	ADMOPUJ	ADMOPUT		
2015	ADMOPUJ	ADMOPUT		
2018	ADMOPUT			
2020	ADMOPMT	ADMOPUJ		
2040	ADMOPRT			
2041	ADMOPRT			
2042	ADMOPRT			
2043	ADMOPRT			
2050	ADM8801			
2051	ADM8801			
2052	ADM8101	ADM8201	ADM8301	ADM8401
	ADM8501	ADM8601	ADM8701	
2053	ADM8101	ADM8201	ADM8301	ADM8401

Msg	Modules that issue the message			
	ADM8501	ADM8601	ADM8701	
2054	ADM8101	ADM8201	ADM8301	ADM8401
	ADM8501	ADM8601	ADM8701	
2055	ADM8101	ADM8201	ADM8301	ADM8401
	ADM8501	ADM8601	ADM8701	
2056	ADM8101	ADM8201	ADM8301	ADM8401
	ADM8501	ADM8601	ADM8701	
2057	ADM8101	ADM8201	ADM8301	ADM8401
	ADM8501	ADM8601	ADM8701	
2058	ADM8101	ADM8201	ADM8301	ADM8401
	ADM8501	ADM8601	ADM8701	
2059	ADM8601			
2060	ADM8DL1			
2061	ADM8101	ADM8202	ADM8203	ADM8301
	ADM8401	ADM8601		
2062	ADM8501			
2063	ADM8501			
2064	ADM8501			
2065	ADM8601			
2066	ADM8201			
2067	ADM8204	ADM8701	ADM8702	
2068	ADM8204	ADM8701		
2069	ADM8204	ADM8701		
2070	ADM8204	ADM8701		
2071	ADM8204	ADM8701	ADM8702	
2072	ADM8204	ADM8601	ADM8701	
2073	ADM8203			
2074	ADM8203			
2075	ADM8101	ADM8201	ADM8301	ADM8401
	ADM8501	ADM8601	ADM8701	
2076	ADM8201	ADM8701	ADM8702	
2104	ADMOPUV			
2105	ADMOPUV			
2106	ADMOPUV			
2110	ADMOPUI			
2111	ADMOPUI			
2120	ADMOPUC			
2121	ADMUPRTC			
2122	ADMUPRTC			
2123	ADMUPRTC			
2124	ADMUPRTC			
2125	ADMUPRTC			
2126	ADMUPRTC			
2127	ADMUPRTC			
2128	ADMUPRTC			
2129	ADMUPRTC			
2130	ADMUPRTC			
2131	ADMUPRTC			
2132	ADMUPRTC			
2133	ADMUPRTC			
2134	ADMUPRTC			
2135	ADMUPRTC			
2138	ADMUPRTC			
2139	ADMUPRTC			
2150	ADMUCDSD			
2151	ADMUCDSD			
2152	ADMUCDSD			
2153	ADMUCDSD			
2154	ADMUCDSD			
2155	ADMUCDSD			
2156	ADMUCDSD			
2200	ADMFOU	ADMFSDU		
2201	ADMFOU	ADMFSDU		
2202	ADMFOU	ADMFSDU		
2203	ADMFSDU			
2206	ADMFSDU			
2207	ADMFOU			
2210	ADMFOU			
2214	ADMFOU			
2400	ADMFOU	ADMFSDU		
2401	ADMFOU	ADMFSDU		
2700	ADM1MSLV			
2701	ADM1MSLV			
2702	ADM1MSLV			
2704	ADM1MSLV			

GDDM messages — 0001 +

Msg	Modules that issue the message			
2705	ADM1MSLV			
2706	ADM1IMDC	ADM1IMDT		
2750	ADMD4PUT			
2751	ADMD4PUT			
2752	ADMD4AEG	ADMD4MEG	ADMD4OEG	ADMD4PUT
2753	ADMD4PUT			
2754	ADMD4PUT			
2755	ADMD4AEG			
2756	ADMD4AEG	ADMD4OEG		
2757	ADMD4MEG			
2758	ADMDLC	ADMDPC	ADMDTC	ADMDWC
	ADM4CDU			
2759	ADMD4QPG			
2760	ADM4CONV			
2761	ADMD4OEG			
2762	ADMD4PUT			
2763	ADMD4OEG			
2764	ADMD4PUT			
2775	ADM4CDU			
2776	ADM4CDU			
2777	ADM4CDU			
2778	ADM4CDU			
2779	ADM4BCM	ADM4CDU	ADM4RER	
2780	ADM4CDU			
2781	ADM4CDI			
2782	ADM4CDG			
2783	ADM4GSF			
2784	ADM4GSF			
2785	ADM4GSF			
2786	ADM4GSF			
2788	ADM4CDU			
2791	ADM4GSF			
2792	ADM4CDI	ADM4CDU		
2795	ADM4CDU			
2850	ADMDDSRD			
2864	ADMDFCG			
2865	ADMDWC			
2866	ADMDWAGD			
2968	ADMDXWIN			
2969	ADMDXCMM			
2970	ADMDXWIN			
2971	ADMDXWIN			
2972	ADMDXWIN			
2973	ADMDXWIN			
2974	ADMDXCPZ			
2975	ADMDXCPZ			
2976	ADMDXCPZ			
2977	ADMDXCPZ			
2979	ADMDXCOU			
2980	ADMDXCOU	ADMDXCSV		
2981	ADMDXCOU			
2983	ADMDXCOU	ADMDXCSV		
2984	ADMDXCSV			
2988	ADMDXCMM	ADMDXCMN		
2989	ADMDXCMN			
2990	ADMDXCMN			
2991	ADMDXCPZ			
2992	ADMDXWIN			
2993	ADMDXWIN			
2994	ADMDXCMC			
2995	ADMDXCMD			
2997	ADMDAPRS			
2998	ADMDXCMC			
2999	ADMDXCMM			
3000	ADMDQDEF			
3001	ADMDQDEF			
3002	ADMDQDEF	ADMDQDEL	ADMDQMOD	ADMDQQRY
	ADMDQQSZ			
3003	ADMDQDEF	ADMDQMOD	ADMDQQRY	
3004	ADMDQVFL	ADMDQVPG		
3005	ADMDQVDB			
3006	ADMDQDEF	ADMDQMOD		
3007	ADMDQMOD			
3008	ADMDQDEL	ADMDQMOD	ADMDQQRY	ADMDQQSZ
3009	ADMDQDEF	ADMDQMOD	ADMDQQRY	

Msg	Modules that issue the message			
3010	ADMDQVBL			
3012	ADMDASCR			
3013	ADMDQDEF	ADMDQMOD		
3100	ADMDBFN1			
3101	ADMDBFN2			
3102	ADMDBFN1	ADMDBFN2		
3103	ADMDBCRT			
3104	ADMDBCRT			
3105	ADMDBFN1			
3106	ADMDBFN1			
3115	ADMDBPN2			
3116	ADMDBPN1			
3117	ADMDQQID			
3118	ADMDBFN1	ADMDBPN1	ADMDBPN2	ADMDEFN1
	ADMDNCRT	ADMDNMOD	ADMDNQRY	ADMDNQWI
	ADMDNQWN	ADMDNQWP	ADMDNSWP	ADMDQQID
	ADMDQQNO			
3119	ADMDBFN1	ADMDBPN2	ADMDEFN1	ADMDNMOD
	ADMDNQRY	ADMDNQWN	ADMDQQNO	
3120	ADMDEFN1			
3121	ADMDBPN1	ADMDEFN2		
3122	ADMDBPN1	ADMDEFN1	ADMDEFN2	
3123	ADMDEFN1			
3124	ADMDEFN1			
3125	ADMDEFN1			
3126	ADMDEFN1			
3127	ADMDEFN1			
3128	ADMDEFN1			
3129	ADMDEFN1			
3130	ADMDEFN1			
3131	ADMDEFN1			
3132	ADMDEFN1			
3133	ADMDEFN1			
3134	ADMDEFN1			
3135	ADMDEFN1			
3150	ADMDCCWIN			
3151	ADMDCCWIN			
3152	ADMDCCWIN			
3153	ADMDCCWIN			
3154	ADMDCCWIN			
3155	ADMDCCPC			
3157	ADMDCCGS	ADMDCCPS		
3158	ADMDCCFW	ADMDCCGS		
3159	ADMDCCENA			
3160	ADMDCCENA			
3161	ADMDRPTX			
3170	ADMDJC	ADMDKC	ADMDLC	ADMDOCO
	ADMDPC	ADMDTC	ADMDWC	
3172	ADMDTC	ADMDWC		
3173	ADMDPC	ADMDWC		
3174	ADMDTPSR			
3175	ADMDWPRL	ADMDXPRD		
3176	ADMDWPIP			
3177	ADMDWPIP			
3178	ADMDKGP	ADMDWUFL		
3179	ADMDJDOC	ADMDLC	ADMD3EMU	
3180	ADMDNCRT	ADMDNDEL	ADMDNMOD	ADMDNQRY
	ADMDNQUN	ADMDNQWI	ADMDNQWN	ADMDNQWP
	ADMDNSEL	ADMDNSWP	ADMDWC	
3181	ADMDNCRT			
3182	ADMDNCRT	ADMDNDEL	ADMDNQWP	ADMDNSEL
	ADMDNSWP			
3183	ADMDNCRT	ADMDNMOD		
3184	ADMDNCRT	ADMDNMOD		
3185	ADMDNCRT	ADMDNMOD		
3186	ADMDNCRT	ADMDNMOD		
3187	ADMDNCRT	ADMDNMOD		
3188	ADMDNCRT	ADMDNMOD		
3189	ADMDNCRT			
3190	ADMDNCRT			
3191	ADMDNCRT	ADMDNMOD	ADMDNQRY	
3192	ADMDNDEL	ADMDNQWP	ADMDNSEL	ADMDNSWP
3193	ADMDBPN1	ADMDNQWP	ADMDNSWP	
3194	ADMDNSWP			
3195	ADMDNDEL			

Msg	Modules that issue the message			
3196	ADMDNQWI			
3200	ADMDGIN			
3201	ADMDGEN	ADMDGIN	ADMDGRQ	
3202	ADMDGEN	ADMDGIN	ADMDGRQ	
3203	ADMDGIN			
3204	ADMUPCT	ADMUPCV	ADMUPGT	ADMUPGV
3206	ADMDGEN	ADMDGIN		
3207	ADMDGPA			
3208	ADMDGEN	ADMDGIN		
3209	ADMDGEN	ADMDGIN		
3210	ADMDGGI			
3211	ADMDGEN			
3212	ADMDGEN	ADMDGIN		
3213	ADMDGEN			
3214	ADMDFPIP	ADMDWPIP		
3215	ADMDWPIP			
3216	ADMDGEN	ADMDGIN		
3217	ADMDGIN			
3218	ADMDGIN			
3219	ADMDGIN			
3220	ADMDGIN			
3221	ADMDGSC			
3222	ADMDGSC			
3223	ADMDGPAX			
3224	ADMDGPC	ADMDGPE	ADMDGPEE	ADMDGPM
	ADMDGPR			
3225	ADMDGPA	ADMDGPAX		
3226	ADMDGSE			
3227	ADMDGSE			
3228	ADMDGSE			
3229	ADMDGSE			
3230	ADMDGPAC			
3231	ADMDGPO			
3232	ADMDGLD			
3233	ADMDGLD			
3234	ADMDGLD			
3235	ADMDGLD			
3236	ADMDGLD			
3237	ADMDGLD			
3238	ADMUPCT	ADMUPCV	ADMUPGT	ADMUPGV
3250	ADMDGXCR			
3251	ADMDGIN			
3252	ADMDGGNS			
3253	ADMDGGNS			
3254	ADMDGPA			
3255	ADMDGWI			
3256	ADMDGWI			
3257	ADMDGWI			
3258	ADMDGPA			
3259	ADMDGPA			
3260	ADMDGGP	ADMDGLD	ADMDGMLD	ADMDGMSV
	ADMDGPA	ADMDGPAC	ADMDGPAD	ADMDGPO
	ADMDGSC	ADMDGSE	ADMDGSV	ADMDGTR
3261	ADMDGPAD			
3262	ADMDGSC			
3263	ADMDGSE			
3264	ADMDGGNS			
3265	ADMDGLD	ADMDGSC		
3266	ADMDGPA			
3267	ADMDGSEN			
3268	ADMDGSC			
3269	ADMUPGT	ADMUPGV		
3270	ADMDGLD			
3271	ADMDGCFU			
3272	ADMDGWI			
3273	ADMDGSC			
3274	ADMDGPD6	ADMDGPT6	ADMDGP3	
3275	ADMDGLD			
3276	ADMDGLD			
3277	ADMDGPD6			
3278	ADMDGI6			
3279	ADMDGSQ			
3280	ADMDKRX	ADMDKVT		
3281	ADMDKDG			
3282	ADMDKDI	ADMDKPC	ADMDKTP	

Msg	Modules that issue the message			
3283	ADMDKDI	ADMDKPC		
3284	ADMDKUT	ADMDKVT		
3285	ADMDKUT	ADMDKVT	ADMOQPU	
3286	ADMDKAP			
3290	ADMDGMLD			
3291	ADMDGMLD	ADMDGMSV		
3292	ADMDGMLD	ADMDGMSV		
3293	ADMDGMA			
3294	ADMDGMLD	ADMDGMSV		
3300	ADM3AAAR	ADM3AARC	ADM3AARL	ADM3AASL
	ADM3AAWN	ADM3AMCP	ADM3AMEX	ADM3AMRP
	ADM3AQQQ	ADM3AQQR	ADM3AQQW	
3301	ADM3AAAR	ADM3AMRP		
3302	ADM3AAAR			
3303	ADM3AALC	ADM3AARS	ADM3AMCP	ADM3AMCV
	ADM3AMER	ADM3AMEX	ADM3AMFR	ADM3AMIV
	ADM3AMMI	ADM3AMOR	ADM3AMRP	ADM3AMSC
	ADM3AMTM	ADM3AQQQ	ADM3AQQS	ADM3AQRQ
	ADM3ATGS	ADM3ATPS		
3304	ADM3AARS			
3305	ADM3AARS			
3306	ADM3AARC	ADM3AARL	ADM3AASL	ADM3AAWN
	ADM3AMCP	ADM3AMEX	ADM3AQQQ	ADM3AQQR
	ADM3AQQW			
3307	ADM3AARC			
3308	ADM3AARC			
3309	ADM3AALC			
3310	ADM3AAMX			
3311	ADM3AAWN			
3312	ADM3AAWN			
3313	ADM3AARS			
3315	ADM3AMCV	ADM3AMSC		
3316	ADM3AMCV	ADM3AMSC		
3317	ADM3AMCV	ADM3AMSC	ADM3ATGS	ADM3ISCL
3318	ADM3AMCV	ADM3AMSC	ADM3ISCL	
3319	ADM3AMOR			
3320	ADM3AMMI			
3321	ADM3ATGS	ADM3ATPS		
3322	ADM3AEEH	ADM3ATGS	ADM3ATPS	
3323	ADM3ATGT			
3324	ADM3ATPE			
3325	ADM3ATGT	ADM3ATPT		
3331	ADM3ATGE	ADM3ATGT		
3332	ADM3ATGE			
3333	ADM3AALC	ADM3AARC	ADM3AAWN	ADM3AMCP
	ADM3AMCV	ADM3AMER	ADM3AMEX	ADM3AMFR
	ADM3AMIV	ADM3AMMI	ADM3AMOR	ADM3AMRP
	ADM3AMSC	ADM3AMTM	ADM3ATGS	
	ADM3AARC	ADM3AAWN	ADM3AMCP	ADM3AMEX
3334	ADM3AARS			
3335	ADM3AARS			
3336	ADM3AMTM			
3337	ADM3AMCP	ADM3AMEX		
3338	ADM3ATGS	ADM3ATPI	ADM3ATPS	
3339	ADM3ATPS			
3340	ADM3ATPS			
3341	ADM3ARO			
3342	ADM3AMCP	ADM3AMCV	ADM3AMEX	ADM3AQRQ
	ADM3ATGS			
3343	ADM3AMCP	ADM3AMEX		
3344	ADM3AASC			
3350	ADMD3SDE	ADM3IACR		
3351	ADMD3SLD	ADM3IAAR	ADM3IACL	ADM3IACR
	ADM3IADE	ADM3IAG	ADM3IAGE	ADM3IAQR
	ADM3IARE	ADM3IATR		
3352	ADMD3SDE	ADM3IACR	ADM3INIT	
3353	ADMD3SDE	ADM3IACR		
3354	ADMD3SDE	ADM3IACR	ADM3INIT	
3355	ADMD3SDE	ADM3SRF	ADM3IAAR	ADM3IACR
	ADM3IRF			
3356	ADMD3DQR	ADMD3SDE	ADMD3SQR	ADMD3SRE
	ADM3IACR	ADM3IAQR	ADM3IARE	ADM3IQRY
	ADM3IRES			
3357	ADMD3SDE	ADMD3SRE	ADM3IACR	ADM3INIT
	ADM3IRES			
3358	ADMD3DIF	ADMD3SDL	ADMD3SLD	ADMD3SQR

GDDM messages — 0001 +

Msg	Modules that issue the message				Msg	Modules that issue the message			
	ADMD3SRE	ADMD3SRF	ADMD3STR	ADMD3SXF	3456	ADMD3DPG			
	ADM3AMCP	ADM3IAAR	ADM3IACL	ADM3IADE	3457	ADMD3DPG			
	ADM3IAQR	ADM3IARE	ADM3IATR		3458	ADMD3DPG			
3359	ADMD3SRF	ADM3IRF			3459	ADMD3DQC	ADMD3DQF		
3360	ADM3IARE	ADM3IPLC	ADM3ISCL	ADM3PROJ	3461	ADMDXRIO	ADMD3DIF		
3361	ADM3IACL	ADM3INIT			3462	ADMD3DIS	ADMD3DPG		
3362	ADMD3CIB	ADMD3DIF	ADMD3STR	ADM3IACL	3463	ADMD3DPL	ADMOQPU		
	ADM3IATR	ADM3ICLR	ADM3ISUB	ADM3ITRM	3470	ADMD3DQR	ADMD3SDE	ADMD3SDL	ADMD3SES
	ADM3PROJ					ADMD3SLD	ADMD3SQR	ADMD3SQS	ADMD3SRE
3363	ADMD3DIF	ADMD3STR	ADM3ICLR	ADM3ITRM		ADMD3SRF	ADMD3STR	ADMD3SXF	
3364	ADM3IASV	ADM3PRST	ADM3PSAV		3471	ADMD3SRF			
3365	ADM3IASV	ADM3PSAV			3472	ADMD3SRF			
3366	ADM3IAGS	ADM3IAXF			3473	ADMD3SQS			
3367	ADM3IAGS	ADM3IAXF			3474	ADMD3SDE	ADMD3SRE		
3368	ADMD3DIF	ADM3IGT	ADM3IGTE	ADM3IGTS	3475	ADMD3SDE			
3369	ADMD3SSD	ADMD3SXF	ADM3ITGT	ADM3PAPF	3476	ADMD3SDE			
	ADM3PAPT	ADM3PSMP			3477	ADMD3SDS			
3370	ADM3IAP	ADM3IAPE	ADM3IAPS	ADM3IAXF	3478	ADMD3SLD	ADMD3SPX	ADMD3SXF	
3371	ADM3IAPS	ADM3IAXF	ADM3IPE2	ADM3IPS2	3479	ADMD3SDS			
	ADM3IPT	ADM3IPTE			3480	ADMD3SDS			
3372	ADMD3SSD	ADMD3SXF	ADM3ITGT		3481	ADMD3SDS			
3373	ADM3IAGS	ADM3IAPS	ADM3IGTS	ADM3IPTS	3482	ADMD3SGS			
3374	ADM3IAGS	ADM3IAPS	ADM3IGTS	ADM3IPTS	3483	ADMD3SLD	ADMD3SPX	ADMD3SXF	
3375	ADM3IAGS	ADM3IAPS			3484	ADMD3SSD	ADMD3SXF		
3376	ADMD3DIF	ADM3IAPS			3490	ADMD3CEN			
3377	ADM3IAPS				3491	ADMD3CIB	ADMD3CIL		
3378	ADMD3DIF	ADMD3SXF	ADM3IAGS	ADM3IAPS	3492	ADMD3CIB			
	ADM3IGTS	ADM3IPS2			3493	ADMD3CIB			
3379	ADMD3DIF	ADM3ATFT	ADM3IAG	ADM3IAP	3494	ADMD3CIB			
	ADM3IGT	ADM3IPT			3495	ADMD3CEN			
3380	ADMD3DIF	ADM3AEEH	ADM3ATBF	ADM3ATFM	3497	ADMD3CIL			
	ADM3ATIF	ADM3ATIM	ADM3ATMF	ADM3ATPI	3498	ADMD3CEN	ADMD3CIL	ADMD3CQL	
	ADM3ATPT	ADM3ATTF	ADM3IASV		3499	ADMD3CEN	ADMD3CIB	ADMD3CQB	
3381	ADMD3DIF	ADM3IAP	ADM3IAPE	ADM3IPE2	3800	ADMJMAIN			
					4000	ADM5IV			
	ADM3IPT	ADM3IPTE			4001	ADM5CUR	ADM5IV	ADM5SHN	
3382	ADMD3DIF	ADM3ATBF	ADM3ATPT	ADM3IPT	4002	ADM5PFK			
3383	ADM3ATPE				4003	ADM5IV			
3384	ADMD3DIF	ADMD3SXF	ADM3IAG	ADM3IAGE	4004	ADM5SHN			
	ADM3IGT	ADM3IGTE			4006	ADM5DC	ADM5ED	ADM5EDD	ADM5EDE
3385	ADM3IAG					ADM5EDF	ADM5EDT	ADM5EX	ADM5IM
3386	ADMDJC	ADMDKC	ADMDLC	ADMDOCO		ADM5IM1	ADM5IP	ADM5LD	ADM5LDR
	ADMDPC	ADMDTC	ADMDWC	ADMD3DIF		ADM5OP	ADM5PJ	ADM5PJD	ADM5PJL
	ADMD3DPG	ADMD3SXF	ADM3IACL	ADM3IATR		ADM5PJS	ADM5PJ1	ADM5PJ2	ADM5PJ3
	ADM3IAXF	ADM3ICLR	ADM3IDEL	ADM3INEG		ADM5PR	ADM5SC	ADM5SCP	ADM5SI
	ADM3IORN	ADM3IPLC	ADM3IQRY	ADM3IREF		ADM5VW			
	ADM3IRES	ADM3ISCL	ADM3ISUB	ADM3ITRM	4007	ADM5EDT	ADM5PJ3		
3387	ADMD3SSD	ADMD3SXF	ADM3ITFR	ADM3PSMP	4009	ADM5PJ1			
3388	ADM3IAGS	ADM3IGTS			4010	ADM5ERR			
3389	ADMD3DIF	ADM3ATPE			4011	ADM5EDC	ADM5EDF	ADM5PJ1	
3390	ADM3IAGS				4012	ADM5EDC	ADM5EDF	ADM5PJ1	
3391	ADM3DCAN				4013	ADM5IM	ADM5IP	ADM5LD	ADM5NMA
3400	ADM3PCRT					ADM5SC			
3401	ADM3IAGS	ADM3IAPS	ADM3IAXF	ADM3PCRT	4014	ADM5EDF	ADM5PR	ADM5VW	
	ADM3PDEL	ADM3PRST	ADM3PSAV	ADM3TREL	4015	ADM5NMA	ADM5PJD	ADM5PJS	
3402	ADMD3SBX	ADM3PDEL	ADM3PSAV	ADM3TREL	4016	ADM5ED	ADM5IM	ADM5IP	ADM5LD
3403	ADMD3DIF	ADMD3SBX	ADMD3SXF	ADM3XFER		ADM5NMA	ADM5SC		
3404	ADM3PROJ				4017	ADM5EX	ADM5IM	ADM5IP	ADM5LD
3405	ADM3PROJ					ADM5NMA	ADM5PJL	ADM5PR	ADM5SC
3406	ADM3PROJ					ADM5SI	ADM5VW		
3407	ADM3PROJ				4018	ADM5ED	ADM5EX	ADM5IM	ADM5IP
3408	ADM3PROJ					ADM5LD	ADM5PJ	ADM5PJC	ADM5PJL
3409	ADM3IPLC	ADM3PROJ				ADM5PR	ADM5SC	ADM5SI	ADM5VW
3410	ADM3ISCL	ADM3PROJ			4019	ADM5DC	ADM5EX	ADM5SI	
3411	ADM3IORN	ADM3PROJ			4020	ADM5ERR	ADM5IM		
3412	ADM3IREF	ADM3PROJ			4021	ADM5ERR	ADM5PR4		
3413	ADM3PROJ				4031	ADM5IV			
3414	ADM3PROJ				4032	ADM5SC			
3415	ADM3PROJ				4034	ADM5SCP			
3450	ADMD3DPG				4035	ADM5MAIN	ADM5NMA		
3451	ADMD3DPG				4036	ADM5IV			
3452	ADMD3DPG				4037	ADM5ED	ADM5IV	ADM5IVP	ADM5NMA
3453	ADMD3CEN	ADMD3DPG	ADMD3DQR	ADMD3SES	4038	ADM5NMA			
3454	ADMD3DPG	ADMD3DPL			4039	ADM5IV	ADM5IVP	ADM5NMA	
3455	ADMD3DPG	ADMD3DQC	ADMD3DQF	ADMD3DQR	4040	ADM5NMA			

Msg	Modules that issue the message			
4041	ADM5PFS			
4042	ADM5PFS			
4043	ADM5NMA	ADM5PFS		
4044	ADM5NMA			
4046	ADM5NMA			
4047	ADM5NMA			
4048	ADM5NMA			
4049	ADM5NMA	ADM5PJ	ADM5PJL	
4050	ADM5NMA			
4051	ADM5NMA			
4053	ADM5SCP			
4054	ADM5SCP			
4061	ADM5EDC	ADM5EDE	ADM5EDF	ADM5EDT
4063	ADM5PJC			
4064	ADM5IP	ADM5IV		
4065	ADM5IV	ADM5OP		
4066	ADM5MAIN	ADM5SHN		
4067	ADM5PFK			
4068	ADM5LDR			
4069	ADM5ED	ADM5EDF	ADM5LD	ADM5NMA
	ADM5PJC	ADM5PR	ADM5SI	ADM5VW
4070	ADM5PJ3			
4072	ADM5CUR	ADM5EDC	ADM5EDE	ADM5EDF
	ADM5EDT			
4073	ADM5EX	ADM5IM		
4074	ADM5PFS			
4077	ADM5NMA			
4078	ADM5PR4			
4079	ADM5PFK			
4081	ADM5PJ	ADM5PJC		
4083	ADM5ERR			
4084	ADM5PJC	ADM5SH		
4085	ADM5SH			
4086	ADM5PFK			
4087	ADM5PJC			
4089	ADM5IM			
4090	ADM5ED	ADM5EDD	ADM5EDE	ADM5EDF
	ADM5EDT			
4091	ADM5PFK			
4092	ADM5ED			
4093	ADM5PR			

GDDM-IMD messages

Msg	Modules that issue the message			
00011	AEMIOS01			
00012	AEMGRP01	AEMGRP02	AEMGRP08	AEMGRP09
	AEMINT00	AEMLIB07	AEMMAP01	AEMMAP02
	AEMMAP08	AEMMAP09	AEMPSG01	AEMUTY00
00014	AEMGRP00	AEMGRP03	AEMGRP06	AEMLIB03
	AEMLIB04	AEMLIB07	AEMMAP00	AEMMAP03
	AEMPSG00	AEMSRV70	AEMTBL00	AEMUTY01
	AEMUTY02	AEMUTY03		
00019	AEMPSG03			
00022	AEMDIA80	AEMGRP03	AEMGRP05	AEMMAP03
	AEMMAP10	AEMTBL01	AEMUTY03	AEMMAP67
	AEMPSG00	AEMPSG05		
00023	AEMGRP00	AEMGRP05	AEMLIB03	AEMLIB04
	AEMLIB07	AEMMAP00	AEMMAP67	AEMPSG00
	AEMTBL00	AEMUTY01	AEMUTY02	
00025	AEMSTG00			
00029	AEMLIB04	AEMLIB07	AEMLIB82	AEMLIB84
	AEMPSG00	AEMUTY01	AEMUTY02	
00030	AEMINT02			
00031	AEMUTY03			
00032	AEMMAP03	AEMPSG00		
00033	AEMINT02			
00034	AEMINT02			
00035	AEMPSG01	AEMPSG05		
00037	AEMMAP67			
00038	AEMPSG95			
00039	AEMMAP90			
00040	AEMUTY01			
00041	AEMTUT80			
00042	AEMTUT82			
00043	AEMTUT81			
00044	AEMIOS05			
00048	AEMUTY01			
00049	AEMMAP28			
00050	AEMMAP28			
00052	AEMGRP03	AEMMAP03	AEMMAP10	AEMMAP56
	AEMMAP67	AEMPSG05		
00053	AEMLIB07			
00055	AEMMAP45			
00057	AEMMAP81			
00058	AEMMAP29	AEMMAP45		
00059	AEMMAP29			
00060	AEMGRP04	AEMGRP08	AEMGRP82	
00063	AEMMAP77	AEMMAP82	AEMMAP90	AEMMAP92
	AEMMAP93			
00065	AEMMAP73			
00066	AEMMAP02	AEMMAP80	AEMPSG00	
00068	AEMMAP02			
00069	AEMGRP09	AEMMAP09		
00079	AEMGRP04			
00080	AEMMSL00			
00081	AEMMSL00			
00082	AEMMSL00	AEMMSL01		
00083	AEMMSL00	AEMMSL02		
00084	AEMMSL00	AEMMSL03	AEMMSL04	AEMMSL05
	AEMMSL06			
00085	AEMMSL03	AEMMSL04	AEMMSL05	AEMMSL06
00086	AEMMSL00	AEMMSL02	AEMMSL83	
00087	AEMGRP00	AEMGRP01	AEMGRP05	AEMGRP82
	AEMGRP83	AEMGRP85	AEMGRP87	AEMMAP02
	AEMMAP04	AEMMAP27	AEMMAP28	AEMMAP29
	AEMMAP80	AEMMAP81	AEMMAP84	AEMMAP87
	AEMMSL77	AEMMSL78	AEMMSL79	AEMMSL87
	AEMPSG00	AEMPSG04	AEMPSG88	AEMPSG99
	AEMTBL01	AEMTBL87	AEMTBL88	
00088	AEMMSL86			
00089	AEMUTY03			
00090	AEMDSP00			
00091	AEMDSP05			
00095	AEMMSL00			
00097	AEMIOS01	AEMMSL00	AEMMSL01	AEMMSL03
	AEMMSL04	AEMMSL05	AEMMSL06	AEMMSL07

GDDM-IMD messages

Msg	Modules that issue the message				Msg	Modules that issue the message			
	AEMMSL77	AEMMSL78	AEMMSL79	AEMMSL82	00171	AEMMAP83			
	AEMMSL84	AEMMSL85	AEMMSL86	AEMMSL87	00172	AEMMAP07	AEMMAP10	AEMPAR80	
	AEMPAR81				00173	AEMMAP83			
00098	AEMMAP10	AEMMAP43	AEMMAP59	AEMMAP60	00174	AEMMAP83			
	AEMMAP66	AEMMAP69			00175	AEMMAP55	AEMMAP74		
00099	AEMGRP00	AEMGRP01	AEMGRP02	AEMGRP81	00176	AEMMAP74			
	AEMMAP00	AEMMAP01	AEMMAP65	AEMMSL77	00177	AEMINT80			
	AEMMSL78	AEMPSG01	AEMSRV80	AEMSRV83	00178	AEMGRP02	AEMGRP83	AEMMAP02	
	AEMUTY01	AEMUTY86			00179	AEMSRV64			
00102	AEMGRP00	AEMMAP00	AEMPSG00		00180	AEMSRV64			
00103	AEMGRP00	AEMMAP00			00183	AEMMAP08			
00104	AEMMAP00				00184	AEMMAP72			
00105	AEMGRP83	AEMMAP00	AEMTBL00		00186	AEMMAP72			
00106	AEMGRP00	AEMMAP00	AEMTBL00		00187	AEMMAP04			
00107	AEMGRP03	AEMGRP05	AEMGRP09	AEMMAP03	00188	AEMMAP04			
	AEMMAP04	AEMMAP05	AEMMAP07	AEMMAP09	00189	AEMMAP78			
	AEMMAP10				00190	AEMGRP00	AEMMAP00		
00109	AEMPMS80				00191	AEMMAP83			
00110	AEMMAP03	AEMMAP83			00192	AEMGRP03	AEMMAP83		
00111	AEMSRV64				00193	AEMMAP67			
00113	AEMPSG73	AEMPSG81	AEMUTY01		00194	AEMGRP03	AEMMAP83		
00115	AEMGRP08	AEMMAP08	AEMTBL01		00195	AEMMAP78			
00116	AEMGRP01	AEMMAP01			00196	AEMSRV64			
00117	AEMMAP61				00199	AEMGRP82			
00118	AEMMAP41	AEMMAP45			00200	AEMGRP82			
00119	AEMMAP41				00201	AEMDIA92			
00120	AEMMAP05				00202	AEMLIB82	AEMPSG00	AEMUTY02	
00121	AEMMAP05				00203	AEMLIB82			
00122	AEMMAP05				00204	AEMINT80			
00123	AEMMAP59				00205	AEMINT80			
00124	AEMMAP81				00206	AEMINT01			
00125	AEMMAP81				00207	AEMMAP90			
00126	AEMMAP42				00208	AEMLIB03			
00127	AEMMAP42				00210	AEMLIB04			
00128	AEMMAP78				00211	AEMLIB03			
00130	AEMPSG99				00212	AEMLIB03			
00131	AEMSRV64				00213	AEMLIB04	AEMLIB82		
00132	AEMSRV64				00214	AEMDIA80	AEMGRP05	AEMMAP90	AEMPAR90
00133	AEMSRV64					AEMTBL01			
00134	AEMSRV64				00215	AEMDIA80			
00135	AEMMAP67	AEMSRV64			00216	AEMDIA80			
00136	AEMSRV63				00217	AEMDIA80			
00137	AEMSRV63				00218	AEMDIA80			
00138	AEMUTY03				00219	AEMDIA80			
00139	AEMUTY03				00220	AEMPSG01			
00141	AEMMAP29	AEMPAR80	AEMSRV90		00224	AEMPSG00			
00142	AEMMAP77	AEMMAP78	AEMMAP82	AEMMAP92	00243	AEMMAP56			
	AEMMAP94	AEMMAP98	AEMPAR80		00244	AEMMAP57			
00143	AEMPSG84				00253	AEMPSG00	AEMUTY01		
00144	AEMMAP29	AEMPAR80			00255	AEMGRP05			
00145	AEMMAP07	AEMMAP55	AEMMAP72	AEMMAP74	00256	AEMMAP90			
	AEMMAP79	AEMMAP95			00258	AEMGRP05			
00146	AEMMAP95	AEMMAP96			00259	AEMPAR90			
00147	AEMMAP78	AEMPAR80	AEMPAR90		00260	AEMPAR90			
00148	AEMMAP07	AEMMAP97			00263	AEMGRP02	AEMMAP02		
00149	AEMMAP04				00271	AEMMAP72			
00150	AEMMAP77	AEMMAP94			00272	AEMMAP98			
00151	AEMMAP77	AEMMAP94			00275	AEMPSG85			
00152	AEMMAP71	AEMPAR80			00276	AEMMAP72			
00153	AEMMAP92				00277	AEMUTY03			
00154	AEMMAP92				00279	AEMGRP02	AEMGRP03		
00155	AEMGRP00	AEMMAP00	AEMTBL00		00280	AEMGRP02	AEMGRP03		
00156	AEMMAP02				00281	AEMGRP02	AEMGRP03		
00157	AEMMAP62				00282	AEMMAP90			
00158	AEMMAP62				00285	AEMMAP83			
00159	AEMMAP77				00286	AEMMAP83			
00160	AEMMAP04	AEMMAP77			00287	AEMGRP82			
00161	AEMMAP76				00288	AEMGRP82			
00162	AEMPSG84				00289	AEMGRP82			
00163	AEMMAP76				00290	AEMGRP82			
00164	AEMMAP78				00291	AEMGRP82			
00165	AEMMAP75				00292	AEMGRP82			
00167	AEMPSG00	AEMPSG05	AEMPSG50		00295	AEMPSG03			
00168	AEMPSG88				00296	AEMTBL01			
00170	AEMMAP98				00297	AEMTBL01			

Msg	Modules that issue the message			
00299	AEMMAP67			
00300	AEMGRP02			
00302	AEMGRP09			
00303	AEMMAP02			
00304	AEMGRP03			
00305	AEMMAP02			
00306	AEMMAP02			
00307	AEMMAP83			
00308	AEMMAP83			
00309	AEMGRP03			
00315	AEMPSG81			
00316	AEMMAP62			
00319	AEMSRV90			
00320	AEMSRV90			
00321	AEMSRV90			
00322	AEMSRV90			
00323	AEMSRV90			
00325	AEMMAP47			
00326	AEMMAP44			
00327	AEMMAP44	AEMMAP90		
00328	AEMMAP44	AEMMAP90		
00330	AEMPSG99			
00332	AEMSRV64			
00333	AEMINT01			
00335	AEMPSG73			
00336	AEMPSG73			
00337	AEMINT01			
00338	AEMMAP10	AEMMAP67		
00339	AEMMAP10	AEMMAP67		
00342	AEMGRP05			
00343	AEMPSG04			
00344	AEMPSG04			
00345	AEMGRP05			
00346	AEMGRP05			
00347	AEMPSG81			
00348	AEMGRP00	AEMMAP00	AEMMAP03	AEMMAP80
	AEMPSG00			
00351	AEMMAP56			
00352	AEMMAP56			
00353	AEMMAP56	AEMMAP67	AEMSRV64	
00354	AEMMAP57			
00355	AEMMAP55	AEMMAP57		
00356	AEMINT80			
00358	AEMMAP67			
00359	AEMMAP54			
00360	AEMMAP54			
00361	AEMMAP54			
00362	AEMMAP54			
00363	AEMMAP54			
00364	AEMMAP54			
00365	AEMMAP54			
00366	AEMMAP54			
00367	AEMMAP54			
00368	AEMMAP07			
00369	AEMSRV75			
00370	AEMPSG04			
00377	AEMPSG01			
00378	AEMPSG82			
00379	AEMPSG81			
00381	AEMMAP62			
00382	AEMMAP02	AEMMAP03	AEMMAP83	
00383	AEMGRP02	AEMGRP03		
00384	AEMMAP62	AEMMAP78		
00385	AEMMAP05	AEMMAP62		
00386	AEMINT01			
00387	AEMMAP55			
00388	AEMMAP55			
00389	AEMMAP55			
00390	AEMMAP55			
00391	AEMMAP55			
00393	AEMPSG04			
00395	AEMPSG84			
00400	AEMMSL00			
00401	AEMMSL00	AEMMSL88		
00402	AEMMSL00			

Msg	Modules that issue the message			
00403	AEMMSL00	AEMMSL03	AEMMSL04	AEMMSL05
	AEMMSL06	AEMMSL86		
00404	AEMMSL88			
00410	AEMPSG05			
00411	AEMPSG05			
00412	AEMPSG50			
00414	AEMMAP29			
00415	AEMMAP29			
00416	AEMMAP08	AEMMAP76		

Appendix D. Trace-string syntax

Product-sensitive programming interface

The syntax of a set of TRCESTR statements is shown, in Backus Naur form, in the table below:

program	→	program_body	
	or	CLEAR program_body	
	or	program_body FORCE	
	or	CLEAR program_body FORCE	
program_body	→	statement	
	or	program_body statement	
statement	→	compound_statement	
	or	simple_statement	
	or	if_statement	
if_statement	→	IF relational_expression THEN statement	
	or	IF relational_expression THEN statement ELSE statement	
compound_statement	→	DO statement_list END	
statement_list	→	statement	
	or	statement_list ; statement	
simple_statement	→	function	
	or	simple_statement function	
relational_expression	→	expression	
	or	expression RELOP expression	
expression	→	term	
	or	expression ADDOP term	
term	→	factor	
	or	term MULOP factor	
factor	→	id	
	or	(relational_expression)	
	or	~factor	
	or	SIGN factor	
id	→	constant	
	or	address	
	or	function	
	or	CHARACTER_STRING	
constant	→	BINARY_NUMBER	
	or	DECIMAL_NUMBER	
	or	HEXADECIMAL_NUMBER	

address	→	expression %
	or	expression GR
	or	expression FR
function	→	VARIABLE_NAME
	or	VARIABLE_NAME (parameter-list)
parameter-list	→	cparameter-item
	or	cparameter-list , parameter-item
parameter-item	→	relational_expression
SIGN	→	+ or –
ADDOP	→	+ or – or OR
MULOP	→	* or / or AND
RELOP	→	EQ or NE or LT or GT or LE or GE

Recognized tokens

The text patterns that match the lexical tokens used by the syntax are defined below using the operators defined in this list:

- &** A & B indicates the character A followed immediately by the character B. For simplicity, this operator is assumed:
 - Between characters that are not operators.
 - After the operator) and before the operator (except that only one & is assumed between each) and (.
- |** A|B indicates either the character A or the character B.
- ()** Parentheses change the relative order of priority of operators.
- { }** Operators enclosed in braces are treated as ordinary characters.
- +** One or more occurrences of the immediately preceding character or group of characters if they are enclosed in parentheses.
- Indicates a range, which may only be defined within:
 - Uppercase letters A through Z
 - Lowercase letters a through z
 - Digits 0 through 9.

Note: Blanks are allowed between the template characters.

Here are some examples of the text patterns that can be described using these operators:

```

a+           matches a aa aaaaa
abc+        matches abc abccccc abcc
(abc)+      matches abc abcabc abcabcabc
(alb)       matches a b
(alb)+      matches a b aba
(alb)+c     matches ac aaaac abac

```

The patterns that correspond to the allowed lexical tokens and their alternative forms are given in this list (uppercase or lowercase characters can be used):

Token	Pattern
+	{+}
-	{-}
OR	{ } (OR)
*	{*}
/	/
AND	{&} (AND)
EQ	= (EQ)
NE	(¬=) (NE)
LT	< (LT)
GT	> (GT)
LE	(<=) (= <) (LE)
GE	(>=) (=>) (GE)
({(}
)	{)}
¬	¬
DO	(DO)
END	(END)
IF	(IF)
THEN	(THEN)
ELSE	(ELSE)
;	;
,	,
GR	(GR)
FR	(FR)
%	%
CLEAR	(CLEAR)
FORCE	(FORCE)
BINARY_NUMBER	B'(0 1)+'
DECIMAL_NUMBER	(0-9)+
HEXADECIMAL_NUMBER	X'(0-9 A-F)+'
CHARACTER_STRING	'(A-Z 0-9 { * { + _ !% .!\$ ¢ { } : ; ? ,)+'
VARIABLE_NAME	(A-Z 0-9)+

The pattern matching to find tokens proceeds according to the following rules:

1. Any character that does not occur in the above table is invalid.
2. A space is automatically added to the end of each line.
3. Matching proceeds from a given start position until:
 - a. An invalid character is reached
 - b. A space is reached

- c. No further match is possible.
- 4. The longest match is chosen.
- 5. If there are two candidates of equal length, the token that appears higher in the above list is chosen.
- 6. After a successful match, the next start position is the next nonblank character.
- 7. If there is no match, the next start position is the first nonblank character *after the next blank*.

Rule 2 means tokens must end at the end of a line.

Rule 3 means that the variable GOTIF is not interpreted as the variable GOT followed by the token IF.

Rule 4 means that \leq is not mistaken as $<$ and $=$, and that FR is not be interpreted as the hexadecimal number F followed by the variable name R.

Rule 5 ensures that THEN is not interpreted as a variable name.

Rule 6 means that tokens that can be distinguished need not be separated by a blank. Thus (1+2) is interpreted as 5 tokens.

Rule 7 means that tokens are not picked out of an invalid string; thus IF is not found in the string xxxxxIFxxx.

Blanks are required to separate tokens that could form a valid single token, otherwise they are ignored.

Note: A consequence of rule 3.b is that blanks cannot appear within a character string. An underscore character within a character string is replaced with a blank after lexical analysis has taken place.

Grammatical constructs that might appear peculiar, for example:

1 fr gr

are considered to be syntactically correct but semantically in error.

_____ End of Product-sensitive programming interface _____

Appendix E. Request Control Parameter codes

General-use programming interface

This appendix lists in numeric order the Request Control Parameter (RCP) codes used by GDDM Base and GDDM-PGF. They are included here to assist you in trace interpretation. They are also used in coding TRCESTR statements. (Refer to “Coding TRCESTR statements” on page 27 for more details.)

Table 1 (Page 1 of 7). GDDM Base RCP codes, in numeric order

Hex	Dec	Call name	Function
00020000	131072	FSTRCE	Control internal trace
00030000	196608	FSEXIT	Specify an error exit, or error threshold, or both
00040000	262144	FSQERR	Query last error
00050000	327680	SPINIT	Initialize GDDM with SPIB
00060000	393216	FSQSYS	Query systems environment
00070000	458752	ESSUDS	Specify source-format user default specification
00080000	524288	ESEUDS	Specify encoded user default specification
000A0000	655360	ESACRT	Create application group
000B0000	720896	ESADEL	Delete application group
000C0000	786432	ESAQRY	Query the current application group
000D0000	851968	ESASEL	Select an application group
000F0000	983040	FSTRAN	Translate character string
00100000	1048576	ESQCPG	Query code page of a GDDM object
00110000	1114112	ESSCPG	Set code page of a GDDM object
00120000	1179648	ESQEUD	Query encoded user default specification
00130000	1245184	ESQUNL	Query length of user-defined nickname information
00140000	1310720	ESQUNS	Query user-defined nickname information
08142000	135536640	ESLIB	Library management
08142400	135537664	ESQOBJ	Query existence of GDDM object on auxiliary storage
081C1000	136056832	ESPCB	Identify program communication block
0C000000	201326592	FSTERM	Terminate GDDM processing
0C000001	201326593	FSINIT	Initialize GDDM processing
0C000002	201326594	FSRNIT	Reinitialize GDDM
0C000200	201327104	DSOPEN	Open a device
0C000201	201327105	DSCLS	Close a device
0C000202	201327106	DSUSE	Specify device usage
0C000203	201327107	DSDROP	Discontinue device usage
0C000204	201327108	DSQUID	Query unique device identifier
0C000205	201327109	DSQUSE	Query device usage
0C000206	201327110	DSQDEV	Query device characteristics
0C000207	201327111	DSRNIT	Reinitialize a device
0C040000	201588736	FSPCRT	Create a page
0C040001	201588737	FSPSEL	Select a page
0C040002	201588738	FSPDEL	Delete a page
0C040003	201588739	FSPCLR	Clear the current page
0C040004	201588740	FSPQRY	Query specified page
0C040005	201588741	FSQCPG	Query current page identifier
0C040006	201588742	MSPQRY	Query current page
0C040100	201588992	SSQF	Query a symbol set on auxiliary storage
0C040101	201588993	PSQSS	Query status of device stores
0C040102	201588994	GSQNSS	Query the number of loaded symbol sets
0C040103	201588995	GSQSS	Query loaded symbol sets
0C040200	201589248	PSLSS	Load a symbol set into a PS store from auxiliary storage

RCP codes

Table 1 (Page 2 of 7). GDDM Base RCP codes, in numeric order

Hex	Dec	Call name	Function
0C040201	201589249	PSLSSC	Conditionally load a symbol set into a PS store from auxiliary storage
0C040202	201589250	PSDSS	Load a symbol set into a PS store from the application program
0C040203	201589251	PSRSV	Reserving or releasing a PS store
0C040300	201589504	GSLSS	Load a graphics symbol set from auxiliary storage
0C040301	201589505	GSDSS	Load a graphics symbol set from the application program
0C040400	201589760	PSRSS	Release a symbol set from a PS store
0C040401	201589761	GSRSS	Release a graphics symbol set
0C040500	201590016	FSQDEV	Query device characteristics
0C040501	201590017	FSQURY	Query device characteristics
0C040900	201591040	FSQUPG	Query unique page identifier
0C040B00	201591552	SSREAD	Read a symbol set from auxiliary storage
0C040B01	201591553	SSWRT	Write a symbol set to auxiliary storage
0C040C00	201591808	FSPWIN	Set page window
0C040C01	201591809	FSQWIN	Query page window
0C040D00	201592064	GSCPG	Set current code page
0C040D01	201592065	GSQCPG	Query code page
0C040E00	201592320	FSENAB	Enable/disable device input
0C080000	201850880	FSALRM	Sound the terminal alarm
0C080100	201851136	ASFCUR	Position the cursor
0C080200	201851392	ASDFLT	Set default field attributes
0C080300	201851648	ASDTRN	Define I/O translation tables
0C080400	201851904	ASFCLR	Clear fields
0C080500	201852160	ASFTYP	Define field type
0C080501	201852161	ASFINT	Define field intensity
0C080502	201852162	ASFCOL	Define field color
0C080503	201852163	ASFPSS	Define primary symbol set for a field
0C080504	201852164	ASFHLT	Define field highlighting
0C080505	201852165	ASFEND	Define field end attribute
0C080506	201852166	ASFOUT	Define output blank-to-null conversion
0C080507	201852167	ASFIN	Define input null-to-blank conversion
0C080508	201852168	ASFTRN	Assign translation table set to a field
0C080509	201852169	ASFTRA	Define field transparency attribute
0C08050A	201852170	ASFSEN	Define field mixed-string attribute
0C08050B	201852171	ASFB DY	Define field outline
0C080600	201852416	ASCHLT	Specify character highlights within a field
0C080601	201852417	ASCCOL	Specify character colors within a field
0C080602	201852418	ASCSS	Specify character symbol sets within a field
0C080603	201852419	ASCPUT	Specify field contents
0C080700	201852672	ASDFLD	Define or delete a single field
0C080800	201852928	ASRFMT	Define multiple fields without deleting existing fields
0C080801	201852929	ASDFMT	Define alphanumeric fields, deleting all existing fields
0C080802	201852930	ASRATT	Define field attributes
0C080900	201853184	ASQHLT	Query character highlights for a field
0C080901	201853185	ASQCOL	Query character colors for a field
0C080902	201853186	ASQSS	Query character symbol sets for a field
0C080903	201853187	ASCGET	Get field contents
0C080A00	201853440	ASQFLD	Query field attributes
0C080B00	201853696	ASQMOD	Query modified fields
0C080C00	201853952	FSREST	Retransmit data
0C080C01	201853953	DSCMF	User Control function
0C080C02	201853954	DSQCMF	Query user control function
0C080D00	201854208	ASMODE	Define the operator reply mode
0C080E00	201854464	ASQMAX	Query the number of fields
0C080E01	201854465	ASQNMF	Query the number of modified fields
0C080F00	201854720	ASQCUR	Query cursor position
0C081100	201855232	ASFMOD	Change field status

Table 1 (Page 3 of 7). GDDM Base RCP codes, in numeric order

Hex	Dec	Call name	Function
0C081300	201855744	ASTYPE	Override alphanumeric character-code assignments
0C081401	201856001	SPMXMP	Control the use of mixed fields by mapping
0C081503	201856259	ASGPUT	Specify double-character field contents
0C081603	201856515	ASGGET	Get double-character field contents
0C081800	201857024	ASQLEN	Query length of field contents
0C0C0000	202113024	GSFLD	Define the graphics field
0C0C0001	202113025	GSPS	Define the picture space
0C0C0002	202113026	GSWIN	Define a graphics window
0C0C0003	202113027	GSVIEW	Define a viewport
0C0C0004	202113028	GSQPS	Query the picture-space definition
0C0C0005	202113029	GSQVIE	Query the current viewport definition
0C0C0006	202113030	GSQWIN	Query the current window definition
0C0C0007	202113031	GSUWIN	Define a uniform graphics window
0C0C000A	202113034	GSQFLD	Query the graphics field
0C0C000B	202113035	GSARCC	Specify aspect-ratio control (for copy)
0C0C000D	202113037	GSBND	Define a data boundary
0C0C000E	202113038	GSQBND	Query the current data boundary definition
0C0C0100	202113280	GSQMAX	Query the number of segments
0C0C0101	202113281	GSQCUR	Query the cursor position
0C0C0102	202113282	GSQSSD	Query symbol set data
0C0C0202	202113538	GSQCEL	Query default graphics cell size
0C0C0203	202113539	GSCLP	Enable and disable clipping
0C0C0204	202113540	GSQCLP	Query the clipping state
0C0C0300	202113792	GSSSEG	Create a segment
0C0C0301	202113793	GSSCLS	Close the current segment
0C0C0302	202113794	GSSDEL	Delete a segment
0C0C0303	202113795	GSCLR	Clear the graphics field
0C0C0309	202113801	GSSATI	Set initial segment attributes
0C0C030A	202113802	GSQATI	Query initial segment attributes
0C0C030B	202113803	GSSATS	Modify segment attributes
0C0C030C	202113804	GSQATS	Query segment attributes
0C0C030D	202113805	GSSPOS	Set segment position
0C0C030E	202113806	GSQPOS	Query segment position
0C0C0311	202113809	GSSORG	Set segment origin
0C0C0312	202113810	GSSPRI	Set segment priority
0C0C0313	202113811	GSQPRI	Query segment priority
0C0C0316	202113814	GSQORG	Query segment origin
0C0C0400	202114048	GSMOVE	Move without drawing
0C0C0401	202114049	GSLINE	Draw a straight line
0C0C0402	202114050	GSPLNE	Draw a series of lines
0C0C0406	202114054	GSMARK	Draw a marker symbol
0C0C0407	202114055	GSMRKS	Draw a series of marker symbols
0C0C0408	202114056	GSAREA	Start a shaded area
0C0C0409	202114057	GSENDA	End a shaded area
0C0C040A	202114058	GSVECM	Vectors
0C0C0500	202114304	GSCHAR	Draw a character string at a specified point
0C0C0501	202114305	GSCHAP	Draw a character string at current position
0C0C0502	202114306	GSQTB	Query the text box
0C0C0600	202114560	GSARC	Draw a circular arc
0C0C0601	202114561	GSELPS	Draw an elliptic arc
0C0C0602	202114562	GSPFLT	Draw a curved fillet
0C0C0700	202114816	GSQCP	Query the current position
0C0C0701	202114817	GSCOL	Set current color
0C0C0702	202114818	GSMIX	Set current foreground color-mixing mode
0C0C0703	202114819	GSLT	Set current line type
0C0C0704	202114820	GSLW	Set current line width

RCP codes

Table 1 (Page 4 of 7). GDDM Base RCP codes, in numeric order

Hex	Dec	Call name	Function
0C0C0705	202114821	GSCM	Set current character mode
0C0C0706	202114822	GSCS	Set current symbol set
0C0C0707	202114823	GSCB	Set character-box size
0C0C0708	202114824	GSCA	Set current character angle
0C0C0709	202114825	GSCD	Set current character direction
0C0C070A	202114826	GSPAT	Set current shading pattern
0C0C070B	202114827	GSMS	Set the current type of marker symbol
0C0C070C	202114828	GSCH	Set current character shear
0C0C070E	202114830	GSFLW	Set current fractional line width
0C0C070F	202114831	GSQFLW	Query the current fractional line width
0C0C0711	202114833	GSQCOL	Query the current color
0C0C0712	202114834	GSQMIX	Query the current color mixing mode
0C0C0713	202114835	GSQLT	Query the current line type
0C0C0714	202114836	GSQLW	Query the current line width
0C0C0715	202114837	GSQCM	Query the current character mode
0C0C0716	202114838	GSQCS	Query the current symbol-set identifier
0C0C0717	202114839	GSQCB	Query character-box size
0C0C0718	202114840	GSQCA	Query character angle
0C0C0719	202114841	GSQCD	Query character direction
0C0C071A	202114842	GSQPAT	Query the current shading pattern
0C0C071B	202114843	GSQMS	Query the current marker symbol
0C0C071C	202114844	GSQCH	Query character shear
0C0C071D	202114845	GSMSC	Set marker scale
0C0C071E	202114846	GSQMSC	Query marker scale
0C0C0900	202115328	GSPUT	Restore graphics data
0C0C0A00	202115584	GSIMG	Draw a graphics image
0C0C0A04	202115588	GSIMGS	Draw a scaled graphics image
0C0C0B00	202115840	GSGETS	Start retrieval of graphics data
0C0C0B01	202115841	GSGETE	End retrieval of graphics data
0C0C0B02	202115842	GSGET	Retrieve graphics data
0C0C0C00	202116096	GSILOC	Initialize locator
0C0C0C01	202116097	GSPIK	Initialize pick device
0C0C0C04	202116100	GSIDVI	Initial data value, integer
0C0C0C05	202116101	GSIDVF	Initial data value, float
0C0C0C06	202116102	GSISTR	Initialize string device
0C0C0C07	202116103	GSISTK	Initialize stroke device
0C0C0C09	202116105	GSQLID	Query logical input device
0C0C0D00	202116352	GSENAB	Enable or disable a logical input device
0C0C0E00	202116608	GSFLSH	Clear the graphics input queue
0C0C0E01	202116609	GSQSIM	Query existence of simultaneous queue entry
0C0C0F00	202116864	GSQCHO	Query choice device data
0C0C0F01	202116865	GSQLOC	Query graphics locator data
0C0C0F02	202116866	GSQPIK	Query pick data
0C0C0F03	202116867	GSQSTR	Query string data
0C0C0F04	202116868	GSQSTK	Query stroke data
0C0C0F05	202116869	GSQPKS	Query pick structure
0C0C1000	202117120	GSTAG	Set current primitive tag
0C0C1001	202117121	GSQTAG	Query current tag
0C0C1102	202117378	GSSAGA	Set all geometric attributes
0C0C1103	202117379	GSSTFM	Set segment transform
0C0C1104	202117380	GSQAGA	Query all geometric attributes
0C0C1105	202117381	GSQTFM	Query segment transform
0C0C1107	202117383	GSSCT	Set current transform
0C0C1200	202117632	GSSAVE	Save a segment
0C0C1201	202117633	GSLOAD	Load segments
0C0C1307	202117895	GSMB	Set marker-box size

Table 1 (Page 5 of 7). GDDM Base RCP codes, in numeric order

Hex	Dec	Call name	Function
0C0C1308	202117896	GSQMB	Query marker box
0C0C130D	202117901	GSTA	Set text alignment
0C0C130E	202117902	GSQTA	Query the current text alignment
0C0C130F	202117903	GSCBS	Set character-box spacing
0C0C1310	202117904	GSQCBS	Query character-box spacing
0C0C1311	202117905	GSAM	Set attribute mode
0C0C1312	202117906	GSQAM	Query the current attribute mode
0C0C1313	202117907	GSPOP	Restore attributes
0C0C1314	202117908	GSSVL	Define segment viewing limits
0C0C1315	202117909	GSQSVL	Query the current segment viewing limits
0C0C1316	202117910	GSQBMX	Query the current background color-mixing mode
0C0C1317	202117911	GSBMIX	Set current background color-mixing mode
0C0C1319	202117913	GSCP	Set current position
0C0C1400	202118144	GSSCPY	Copy a segment
0C0C1401	202118145	GSSINC	Include a segment
0C0C1402	202118146	GSCALL	Call a segment
0C0C1500	202118400	GSCORR	Explicit correlation of tag to primitive
0C0C1501	202118401	GSCORS	Explicit correlation of structure
0C0C1900	202119424	GSDEFS	Start the drawing defaults definition
0C0C1901	202119425	GSDEFE	End drawing defaults definition
0C0C1A00	202119680	FSUPDM	Set update mode
0C0C1A01	202119681	FSQUPD	Query update mode
0C0C1B00	202119936	GSEN	Set mixed string attribute of graphics text
0C0C1B01	202119937	GSQSEN	Query mixed string attribute of graphics text
0C0C1F00	202120960	CGLOAD	Load a picture from a Computer Graphics Metafile (CGM)
0C0C2000	202121216	CGSAVE	Save segments in a Computer Graphics Metafile (CGM)
0C100000	202375168	ASREAD	Device output/input
0C100001	202375169	FSFRCE	Update the display
0C100002	202375170	FSCHEK	Check picture complexity before output
0C100003	202375171	GSREAD	Await graphics input
0C100004	202375172	FSSAVE	Save current page contents
0C100005	202375173	FSSHOW	Display a saved picture
0C100007	202375175	FSSHOR	Extended FSSHOW
0C100008	202375176	WSIO	Windowed device input/output
0C10000C	202375180	DSFRCE	Output member to a PDS
0C180000	202899456	FSOPEN	Open alternate device
0C180001	202899457	FSCOPY	Send page to alternate device
0C180002	202899458	GSCOPY	Send graphics to alternate device
0C180003	202899459	FSLOG	Send character string to alternate device
0C180004	202899460	FSCLS	Close alternate device
0C180005	202899461	FSLOGC	Send character string with carriage-control character to alternate device
0C180008	202899464	DSCOPY	Send transformed picture to alternate device
0C200000	203423744	PTSCRT	Create a partition set
0C200001	203423745	PTSQRY	Query partition set attributes
0C200100	203424000	PTSSEL	Select a partition set
0C200101	203424001	PTSDEL	Delete a partition set
0C200102	203424002	PTSQUN	Query unique partition set identifier
0C200300	203424512	PTSSPP	Set partition viewing priorities
0C200301	203424513	PTSQPP	Query partition viewing priorities
0C200400	203424768	PTSQPI	Query partition identifiers
0C200401	203424769	PTSQPN	Query partition numbers
0C240000	203685888	PTNCRT	Create a partition
0C240001	203685889	PTNQRY	Query the current partition
0C240002	203685890	PTNMOD	Modify the current partition
0C240100	203686144	PTNSEL	Select a partition
0C240101	203686145	PTNDEL	Delete a partition

RCP codes

Table 1 (Page 6 of 7). GDDM Base RCP codes, in numeric order

Hex	Dec	Call name	Function
0C240102	203686146	PTNQUN	Query unique partition identifier
0C280000	203948032	MSREAD	Present mapped data
0C280100	203948288	MSPCRT	Create a page for mapping
0C280300	203948800	MSQGRP	Query mapgroup characteristics
0C280301	203948801	MSQMAP	Query map characteristics
0C280302	203948802	MSQADS	Query application data structure definition
0C280303	203948803	MSQFIT	Query map fit
0C280400	203949056	MSQMOD	Query modified fields
0C280500	203949312	MSDFLD	Create or delete a mapped field
0C280501	203949313	MSPUT	Place data into a mapped field
0C280502	203949314	MSGET	Retrieve data from a map
0C280503	203949315	MSQFLD	Query mapped field characteristics
0C280600	203949568	MSCPOS	Set cursor position
0C280601	203949569	MSQPOS	Query cursor position
0C2C0000	204210176	WSCRT	Create an operator window
0C2C0100	204210432	WSDDEL	Delete operator window
0C2C0200	204210688	WSMOD	Modify the current operator window
0C2C0300	204210944	WSQRY	Query the current operator window
0C2C0400	204211200	WSQUN	Query unique operator window identifier
0C2C0500	204211456	WSQWI	Query operator window identifiers
0C2C0600	204211712	WSQWN	Query operator window numbers
0C2C0700	204211968	WSQWP	Query operator window viewing priorities
0C2C0800	204212224	WSSEL	Select an operator window
0C2C0900	204212480	WSSWP	Set operator window viewing priorities
0C300000	204472320	ISFLD	Define image field
0C300001	204472321	ISQFLD	Query image field
0C300002	204472322	ISCTL	Set image quality-control parameters
0C300003	204472323	ISXCTL	Extended set image quality control parameters
0C300B00	204475136	ISESCA	Control echoing of scanner image
0C300C00	204475392	ISLDE	Load external read-only image
0C300D00	204475648	ISQSCA	Query image scanner device
0C300E00	204475904	ISQRES	Query supported image resolutions
0C301200	204476928	ISENAB	Enable or disable image cursor
0C301300	204477184	ISQLOC	Query image locator cursor position
0C301400	204477440	ISILOC	Initialize image locator cursor
0C301500	204477696	ISQBOX	Query image box cursor
0C301600	204477952	ISIBOX	Initialize image box cursor
0C301700	204478208	ISQFOR	Query image formats supported by the device
0C301800	204478464	ISQCOM	Query image compressions supported by the device
0C380000	204996608	APDEF	Define a field list
0C380100	204996864	APDEL	Delete a field list
0C380200	204997120	APMOD	Modify a field list
0C380300	204997376	APQIDS	Query field list identifiers
0C380400	204997632	APQNUM	Query field list numbers
0C380500	204997888	APQRY	Query a field list
0C380600	204998144	APQSI	Query a field list size
0C380700	204998400	APQID	Query unique field list identifier
3C010001	1006698497	IMACRT	Create an image
3C010002	1006698498	IMAGID	Get and reserve a unique image identifier
3C010004	1006698500	IMAQRY	Query attributes of an image
3C010006	1006698502	IMARES	Convert the resolution attributes of an image
3C010007	1006698503	IMADEL	Delete the image associated with the identifier
3C010008	1006698504	IMACLR	Clear a rectangle in an image
3C010009	1006698505	IMATRM	Trim an image down to the specified rectangle
3C01000A	1006698506	IMASAV	Save image on auxiliary storage
3C01000B	1006698507	IMARST	Restore image from auxiliary storage

Table 1 (Page 7 of 7). GDDM Base RCP codes, in numeric order

Hex	Dec	Call name	Function
3C01000C	1006698508	IMARF	Change resolution flag of an image
3C010011	1006698513	IMAPTS	Start data entry into an image
3C010012	1006698514	IMAPT	Enter data into an image
3C010013	1006698515	IMAPTE	End data entry into an image
3C010014	1006698516	IMAGTS	Start retrieval of data from an image
3C010015	1006698517	IMAGT	Retrieve image data from an image
3C010016	1006698518	IMAGTE	End retrieval of data from an image
3C010017	1006698519	IMXFER	Transfer data between two images, applying a projection
3C030001	1006829569	IMPGID	Get and reserve a unique projection identifier
3C030003	1006829571	IMPCRT	Create an empty projection
3C030004	1006829572	IMPDEL	Delete projection
3C030005	1006829573	IMPSAV	Save projection on auxiliary storage
3C030006	1006829574	IMPRST	Restore projection from auxiliary storage
3C030101	1006829825	IMREX	Define rectangular sub-image in pixel coordinates
3C030102	1006829826	IMREXR	Define rectangular sub-image in real coordinates
3C030103	1006829827	IMRPL	Define place position in pixel coordinates
3C030105	1006829829	IMRSCL	Scale extracted image
3C030106	1006829830	IMRRAL	Set current resolution/scaling algorithm
3C030107	1006829831	IMRORN	Turn an extracted image clockwise through a number of right angles
3C030108	1006829832	IMRREF	Reflect extracted image
3C030109	1006829833	IMRNEG	Negate the pixels of an extracted image
3C030201	1006830081	IMRCVB	Define bi-level conversion algorithm
3C030202	1006830082	IMRBRI	Define brightness conversion algorithm
3C030203	1006830083	IMRCON	Define contrast conversion algorithm
3C030204	1006830084	IMRPLR	Define place position in real coordinates
40000000	1073741824	CDPU	Control the printing of Composite Documents

Table 2 (Page 1 of 3). GDDM-PGF RCP codes, in numeric order

Hex	Dec	Call name	Function
10000100	268435712	CHTERM	Terminate the PG routines
10010100	268501248	CHRNIT	Reinitialize PG routines
10020101	268566785	CHSET	Specify chart options
10020201	268567041	CHKEY	Legend key labels
10020202	268567042	CHHEAD	Heading text
10020301	268567297	CHMARK	Component marker table
10020302	268567298	CHLT	Component line type table
10020303	268567299	CHCOL	Component basic color table
10020304	268567300	CHPAT	Component shading pattern table
10020305	268567301	CHLW	Component line width table
10020306	268567302	CHPEXP	Exploded slices in pie charts
10020307	268567303	CHLC	Component line color table
10020401	268567553	CHXSET	X-axis options
10020402	268567554	CHYSET	Y-axis options
10020403	268567555	CHZSET	Z-axis options
10020501	268567809	CHXTTL	X-axis title specification
10020502	268567810	CHYTTL	Y-axis title specification
10020503	268567811	CHXLAB	X-axis label text
10020504	268567812	CHYLAB	Y-axis label text
10020505	268567813	CHXDLB	X-axis data labels
10020507	268567815	CHZDLB	Z-axis data labels
10020607	268568071	CHXSCL	X-axis scale factor
10020608	268568072	CHYSCL	Y-axis scale factor
10020609	268568073	CHXMTH	X-axis month labels

RCP codes

Table 2 (Page 2 of 3). GDDM-PGF RCP codes, in numeric order

Hex	Dec	Call name	Function
1002060A	268568074	CHYMT	Y-axis month labels
1002060B	268568075	CHXDAY	X-axis day labels
1002060C	268568076	CHYDAY	Y-axis day labels
1002060F	268568079	CHNUM	Set number of components
10020610	268568080	CHGAP	Spacing between bars
10020611	268568081	CHGGAP	Spacing between bar groups
10020612	268568082	CHHMAR	Horizontal margins
10020613	268568083	CHVMAR	Vertical margins
10020614	268568084	CHPIER	Reduce pie chart size
10020615	268568085	CHCGRD	Basic character spacing/size
10020616	268568086	CHVCHR	Number of characters in bar values
1002061A	268568090	CHFINE	Curve fitting smoothness
1002061B	268568091	CHZGAP	Spacing between towers
10020701	268568321	CHAATT	Axis line attributes
10020702	268568322	CHGATT	Grid line attributes
10020801	268568577	CHKEYP	Legend base position
10020901	268568833	CHHATT	Heading text attributes
10020902	268568834	CHTATT	Axis title text attributes
10020903	268568835	CHLATT	Axis label text attributes
10020905	268568837	CHKATT	Legend text attributes
10020906	268568838	CHVATT	Attributes of values text in bar and pie charts
10020907	268568839	CHXLAT	X-axis label attributes
10020908	268568840	CHYLAT	Y-axis label attributes
10020909	268568841	CHXTAT	X-axis title attributes
1002090A	268568842	CHYTAT	Y-axis title attributes
1002090B	268568843	CHZLAT	Z-axis label attributes
10020A02	268569090	CHAREA	Chart area
10030C01	268635137	CHMKSC	Set marker scale values
100A0601	269092353	CHXRNG	X-axis explicit range
100A0602	269092354	CHYRNG	Y-axis explicit range
100A0603	269092355	CHXINT	X-axis interception point
100A0604	269092356	CHYINT	Y-axis interception point
100A0605	269092357	CHXTIC	X-axis scale mark interval
100A0606	269092358	CHYTIC	Y-axis scale mark interval
100A0618	269092376	CHKOFF	Legend offsets
100A0619	269092377	CHKMAX	Maximum legend width/height
100A061C	269092380	CHZTIC	Z-axis scale mark interval
100A061D	269092381	CHZRNG	Z-axis explicit range
100B0617	269157911	CHNOFF	Specify offsets for CHNOTE
100B0703	269158147	CHSSEG	Set segment number
100B0904	269158660	CHNATT	Specify attributes for notes
100B0A01	269158913	CHDATT	Datum line attributes
100B0A03	269158915	CHBATT	Set framing box attributes
100B0A04	269158916	CHVDIG	Set decimal digits for bars and tables
100B0A05	269158917	CHTHRS	Bar value threshold limit
100D0A01	269289985	CHBAR	Plot a bar chart
100D0A02	269289986	CHHIST	Histograms
100D0A03	269289987	CHPLOT	Line graphs and scatter plots
100D0A04	269289988	CHSURF	Surface charts
100D0A05	269289989	CHVENN	Venn diagram
100D0A06	269289990	CHPIE	Pie charts
100D0A07	269289991	CHBARX	Plot a bar chart with numeric x-axis values
100D0A08	269289992	CHPOLR	Plot a polar chart
100D0A09	269289993	CHTOWR	Plot a tower chart
100D0A0B	269289995	CHDTAB	Construct a table chart
100E060D	269354509	CHXDTM	X-axis datum line

Table 2 (Page 3 of 3). GDDM-PGF RCP codes, in numeric order

Hex	Dec	Call name	Function
100E060E	269354510	CHYDTM	Y-axis datum line
100F0801	269420545	CHXSEL	X-axis selection
100F0802	269420546	CHYSEL	Y-axis selection
10110100	269549824	CHSTRT	Reset the processing state to state-1
10130100	269680896	CHNOTE	Construct a character string at a designated position
10140100	269746432	CHDRAX	Specific control of axis drawing
10150101	269811969	CHPCTL	Control pie chart slices
10150201	269812225	CHDCTL	Control the format of values, and the overall size of table charts
10150301	269812481	CHMISS	Missing values on a table chart
10160100	269877504	CHTPRJ	Tower chart projection
10170100	269943040	CHCONV	Convert coordinate values
10170200	269943296	CHQPOS	Query positional information
10170300	269943552	CHQRNG	Query axis ranges
10180100	270008576	CHQARE	Query chart area
14000000	335544320	CHART	Invoke Interactive Chart Utility
14000004	335544324	CSSICU	Start an ICU session for a chart
14000010	335544336	CSLOAD	Restore saved chart information
14000014	335544340	CSSAVE	Save chart information
14000020	335544352	CSDIR	Build object directory list
14000024	335544356	CSQDIR	Query object directory list
14000824	335546404	CSDEL	Delete item for a chart
14040000	335806464	CSCCRT	Create a chart
14040004	335806468	CSCDEL	Delete a chart
14040010	335806480	CSQUID	Query unique chart identifier
14040014	335806484	CSQCS	Query CSxxxx call information
14080000	336068608	CSINT	Set integer values for a chart
14080004	336068612	CSFLT	Set floating-point values for a chart
14080008	336068616	CSCHA	Set character values for a chart
14080010	336068624	CSXDT	Set independent (x) data values for a chart
14080014	336068628	CSYDT	Set dependent (y) data values for a chart
14080018	336068632	CSZDT	Set data group (z) data values for a chart
14080020	336068640	CSNUM	Set control value for a chart
14080030	336068656	CSXSL	Set independent (x) data selection for a chart
14080038	336068664	CSZSL	Set data group (z) selection for a chart
140C0000	336330752	CSQINT	Query integer values for a chart
140C0004	336330756	CSQFLT	Query floating-point values for a chart
140C0008	336330760	CSQCHA	Query character values for a chart
140C000C	336330764	CSQCHL	Query character lengths for a chart
140C0010	336330768	CSQXDT	Query independent (x) data values for a chart
140C0014	336330772	CSQYDT	Query dependent (y) data values for a chart
140C0018	336330776	CSQZDT	Query data group (z) values for a chart
140C0020	336330784	CSQNUM	Query control value for chart
140C0030	336330800	CSQXSL	Query independent (x) data selection for a chart
140C0038	336330808	CSQZSL	Query data group (z) selection for a chart

End of General-use programming interface

Glossary

This glossary defines technical terms used in GDDM documentation. If you do not find the term you are looking for, refer to the index of the appropriate GDDM manual or view the *IBM Dictionary of Computing*, located on the Internet at:

<http://www.networking.ibm.com/nsg/nsgmain.htm>

A

AAB. Application anchor block.

ACB. Application control block.

active operator window. In GDDM, the operator window with the highest priority in the viewing order.

active partition. The partition containing the cursor. Contrast with *current partition*.

advanced function printing. The ability of licensed programs to use the all-points-addressable concept to print text and illustrations.

adjunct. In mapped alphanumerics, one of a set of optional subfields in an application data structure that specifies some attribute of a data field; for example, that it is highlighted. An adjunct enables the attribute to be varied at run time.

ADMGDF. See *graphics data format (GDF)*.

ADS. Application data structure.

AFPDS. Advanced-function presentation data stream.

AIC. Application interface component.

alphanumeric character attributes. In GDDM, the highlighting, color, and symbol set to be used for individual characters.

alphanumeric cursor. A physical indicator on a display. It can be moved from one hardware cell to another.

alphanumeric field. A field (area of a screen or printer page) that can contain alphabetic, numeric, or special characters. In GDDM, contrast with *graphics field*.

alphanumeric field attributes. In GDDM, the intensity, highlighting, color, and symbol set to be used for field type, field end, output conversion, input conversion, translate table assignment, transparency, field outlining, and mixed-string fields.

alphanumerics. Pertaining to alphanumeric fields. In GDDM there are three types of alphanumerics:

- Procedural alphanumerics
- Mapped alphanumerics
- High performance alphanumerics (HPA)

alternate device. In GDDM, a device to which copies of the primary device's output are sent. Usually the alternate device is a printer or plotter. See also *primary device*.

annotation. An added descriptive comment or explanatory note.

APA. All points addressable.

aperture. See *pick aperture*.

API. Application programming interface.

APL. One of the programming languages supported by GDDM.

application data structure (ADS). A structure created by GDDM-IMD that contains an entry for each variable field within a *map*. The data to be displayed in a mapped field is placed into the application data structure by the user's program.

application image. In GDDM, an image contained in GDDM main storage, and independent of any device or GDDM page. Contrast with *device image*.

application programming interface (API). The formally defined interface used by an application programmer to pass commands to, and get responses from, an IBM system control program or licensed program.

area. In GDDM, a shaded shape, such as a solid rectangle. It is created by opening the area, defining its outline, and closing the area.

aspect ratio. The width-to-height ratio of an area, symbol, or shape.

attention identifier. A number indicating which button the operator pressed to satisfy a read operation. For example, 0 (returned from GDDM to the application program) means that the operator pressed the Enter key.

attribute byte. The screen position that precedes an alphanumeric field on a 3270-family device and holds the attribute information. See also *trailing attribute byte*.

glossary

attributes. Characteristics or properties that can be controlled, usually to obtain a required appearance; for example, the color of a line. See also *alphanumeric character attributes*, *alphanumeric field attributes*, and *graphics attributes*.

axis. In a chart, a line that is drawn to indicate units of measurement against which items in the chart can be viewed.

A3. A paper size, more common in Europe than in the U.S. It measures 297mm by 420mm, and is twice the size of A4. See also A4.

A4. A paper size, more common in Europe than in the U.S. It measures 210mm by 297mm, and is half the size of A3. Compare with *quarto*. See also A3.

B

background color. Black on a display, white on a printer. The initial color of the display medium. Contrast with *neutral color*.

bar code. A code representing characters by sets of vertical parallel bars of varying thickness and separation that are read optically by transverse scanning.

BASIC. One of the programming languages supported by GDDM.

BDAM. Basic Direct Access Method.

bi-level image. An image in which each pixel is either black or white (value 0 or 1). Contrast with *gray-scale image* and *halftone image*.

BMS. Basic Mapping Support (CICS).

BPAM. Basic Partitioned Access Method.

business graphics. The methods and techniques for presenting commercial and administrative information in chart form; for example, the creation and display of a sales bar chart. Contrast with *general graphics*.

C

CALS. Continuous Acquisition and Life-Cycle Support.

CDPDS. Composite Document Presentation Data Stream.

CDPF. Composed Document Print Facility.

CDPU. Composite Document Print Utility.

CECP. Country-extended code page.

cell. See *character cell*.

CGM. Computer Graphics Metafile. A file that contains information about the content of a picture, and conforms to the International Standard, ISO 8632, or is of a similar format.

channel-attached. Pertaining to devices that are attached directly to a computer by means of data (I/O) channels. Synonymous with *local*. Contrast with *link-attached*.

character. A letter, digit, or other symbol.

character attributes. See *alphanumeric character attributes*. See also *graphics text attributes*.

character box. In GDDM, the rectangle or (for sheared characters) the parallelogram boundaries that govern the size, orientation, spacing, and italicizing of individual symbols or characters to be shown on a display screen or printer page.

The box width, height, and, if required, shear are specified in world coordinates and can be program-controlled. See also *character mode*. Contrast with *character cell*.

character cell. The physical, rectangular space in which any single character or symbol is displayed on a screen or printer device. The size and position of a character cell are fixed. Size is usually specified in pixels on a given device; for example, 9 by 12 on an IBM 3279 Model 3 display. Position is addressed by row and column coordinates. Synonymous with *hardware cell* and *symbol cell*. Contrast with *character box*.

character code. The means of addressing a symbol in a symbol set, sometimes called *code point*.

The particular form and range of codes depends on the GDDM context. For example:

- For the Image Symbol Editor, a hexadecimal constant in the range X'41' through X'FE', or its EBCDIC character equivalent
- For the Vector Symbol Editor, a hexadecimal constant in the range X'00' through X'FF', or its EBCDIC character equivalent
- For the GDDM API, a decimal constant in the range 0 through 239, or subsets of this range (for example, a marker symbol code range of 1 through 8)

character grid. A notional grid that covers the *graphics field*. The size of the grid determines the basic size of the characters in all text constructed by presentation graphics routines. It is the fundamental measurement in chart layout, governing the spacing of mode-2 characters and the size of mode-3 characters. It also governs the size of the chart margins and thus the plotting area.

character matrix. Synonym for *dot matrix*.

character mode. In GDDM, the type of characters to be used. There are three modes:

- Mode-1 characters are loadable into PS and are of device-dependent fixed size, spacing, and orientation, as are hardware characters.
- Mode-2 characters are image (ISS) characters. Size and orientation are fixed. Spacing is variable by program.
- Mode-3 characters are vector (VSS) characters. Box size, position, spacing, orientation, and shear of individual characters are variable by program.

chart. In GDDM, usually means business chart; for example, a *bar chart*.

choice device. A logical input device that enables the application program to identify keys pressed by the terminal operator.

CICS. Customer Information Control System. A subsystem of MVS or VSE under which GDDM can be used.

clipping. In computer graphics, removing parts of a display image that lie outside a viewport. Synonymous with *scissoring*.

CMS. Conversational Monitor System. A time-sharing subsystem that runs under VM/SP.

COBOL. One of the programming languages supported by GDDM.

code page. Defines the relationship between a set of code points and graphic characters. This relationship covers both the standard alphanumeric characters and the national language variations. GDDM supports a set of code pages used with typographic fonts for the IBM 4250 page printer.

code point. Synonym for *character code*.

Composite Document Presentation Data Stream (CDPDS). A data stream containing graphics, image, and text that is the input to the GDDM Composite Document Print Utility (CDPU).

Composed Document Print Facility (CDPF). An IBM licensed program for processing documents destined for the IBM 4250 page printer.

composed-page image file. An intermediate form, residing on disk, of a picture destined for a page printer.

composed-page printer. See *page printer*.

composed-page printer format. A general term describing the format of print data destined for output by using either *CDPF* or *PSF*.

composite document. A document that contains both formatted text, such as that produced by the DCF program, and graphic or image data, such as that produced by GDDM. It is a combination of text and pictures on a page or set of pages. The pictures can be computer graphics or images created by scanning paper originals.

Composite Document Print Utility (CDPU). A utility that can print or display composite documents

compressed data stream. A data stream that has been made more compact by use of a data-compression algorithm.

constant data. In GDDM, data that is defined in a map and need not be known to the application program.

correlation. The translation (by GDDM) of a screen position into a part of the user's picture. This follows a *pick* operation.

country-extended code page (CECP). An extension of a normal EBCDIC code page that includes definitions of all code points in the range X'41' through X'FE'. Each code page contains the same 190 characters, but the mapping between code points and graphics characters depends on the country for which the code page is defined. This is a method of marking a GDDM object so that the environment in which it was created can be identified. It enables automatic translation to a different environment.

CSD. (1) Under MVS or VSE, CICS system definition. (2) In personal computer systems, Corrective Service Diskette; the means by which service is applied to the personal computer system.

current partition. The partition selected for processing by the application program. Contrast with *active partition*.

current position. In GDDM, the end of the previously drawn primitive. Unless a "move" is performed, this position is also the start of the next primitive.

cursor. A physical indicator that can be moved around a display screen. See *alphanumeric cursor* and *graphics cursor*.

CUT. Control unit terminal.

D

DASD. Direct access storage device.

data stream compatibility (DSC). In IBM 8100 systems, the facility that provides access to System/370 applications that communicate with IBM 3270 Information Display System terminals.

data stream compression. The shortening of an I/O data stream for the purpose of more efficient transmission between link-attached units.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

DBCS. Double-byte character set.

DCF. Document Composition Facility.

DCSS. Discontiguous saved segment (VM/SP).

DCT. Destination control table (CICS).

default value. The value of an attribute chosen by GDDM when no value is explicitly specified by the user. For example, the default line type is a solid line. The default value is sometimes device-dependent. See also *drawing default* and *standard default*.

denibblized data. The decoded data stream used between the GDDM DOS Support feature in the host and GDDM-PCLK on the workstation.

designator character. The first byte of a light-pen-detectable field that indicates whether or not the field has been selected.

device echo. A visual identification of the position of the graphics cursor. The form of the device echo is defined by the application program.

device family. In GDDM, a device classification that governs the general way in which I/O will be processed. See also *processing option*. For example:

- Family 1: 3270 display or printer
- Family 2: queued printer
- Family 3: system printer (alphanumerics only)
- Family 4: page printer

device image. In GDDM, an image contained in a device or GDDM page. Contrast with *application image*.

device suffix. In GDDM-IMD, a suffix to a mapgroup name that indicates the device class.

device token. In GDDM, an 8-byte code giving entry to a table of pre-established device hardware characteristics that are required when the device is opened (initialized).

DIF. In GDDM terms, data interchange format.

digital image. A two-dimensional array of picture elements (pixels) representing a picture. A digital image can be stored and processed by a computer, using bits to represent pixels. In GDDM, pixels have the value black or white. Often called simply *image*.

direct transmission. In GDDM image processing, the transfer of image data direct from a source outside GDDM to an image device, including manipulation by a projection in the device, and without GDDM maintaining a copy or buffer of the data.

display device. Any output unit that gives a visual representation of data; for example, a screen or printer. More commonly, the term is used to mean a screen and not a printer.

display point. Synonym for *pixel*.

display-point matrix. Synonym for *dot matrix*.

display terminal. An input/output unit by which a user communicates with a data processing system or subsystem. It usually includes a keyboard and always provides a visual presentation of data. For example, an IBM 3179 display.

DL/1. Data language 1. A language for database processing operations.

dot matrix. In computer graphics, a two-dimensional pattern of dots used for constructing a display image. This type of matrix can be used to represent characters by dots. Synonymous with *character matrix* and *display-point matrix*.

double-byte characters. See *double-byte character set (DBCS)*.

double-byte character set (DBCS). A set of characters in which each character occupies two byte positions in internal storage and in display buffers. Used for oriental languages; for example, *Kanji* or *Hangeul*. Contrast with *single-byte character set*.

DPCX. Distributed Processing Control Executive. An IBM 8100 system control program.

DPPX. Distributed Processing Programming Executive. An IBM 8100 system control program.

drawing default. The value of a graphics attribute chosen by GDDM when no value is explicitly specified

by the user. The drawing default may be altered by the user.

DSC. Data stream compatibility.

dual characters. See *double-byte characters*.

dummy device. An output destination for which GDDM does all the normal processing but for which no actual output is generated. Used, for example, to test programming for an unavailable output device.

E

EBCDIC. Extended binary coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

echo. In interactive graphics, the visible form of the locator or other logical input device.

ECSA. Extended character set adapter.

edit. To enter, modify, or delete data.

editing grid. In the GDDM Image and Vector Symbol Editors, a grid used as a guide for editing a symbol. In the Image Symbol Editor, it is a dot matrix. In the Vector Symbol Editor, it is a grid of lines.

enterprise. An organization or company that undertakes local, national, or international business ventures.

extended data stream. For IBM 3179, 3192, 3278, 3279, and 3287 devices, input/output data formatted and encoded in support of color, programmed symbols, and extended highlighting. These features extend the IBM 3270 data stream architecture.

extended highlighting. The emphasizing of a displayed character's appearance by blinking, underscore, or reverse video.

external defaults. GDDM-supplied values that users can change to suit their own needs.

extracted image. In GDDM, an image on which transform element calls operate. It may imply the whole source image or just a part of it, depending on whether a define sub-image transform element has been applied in its derivation.

F

FCT. File control table (CICS).

field. An area on the screen or the printed or plotted page. See *alphanumeric field*, *graphics field*, and *mapped field*.

field attributes. See *alphanumeric field attributes*.

field list. The high performance alphanumerics data structure used to define alphanumeric fields.

fillet. A curve that is tangential to the end points of two adjoining lines.

flat file. A file that contains only data; that is, a file that is not part of a hierarchical data structure. A flat file can contain fixed-length or variable-length records.

floating area. The part of a page reserved for *floating maps*.

floating map. A map whose absolute position on the GDDM page is not fixed. During execution, a floating map takes the next available space that satisfies its specification.

floating-point feature. A processing unit feature that provides four 64-bit floating-point registers to perform floating-point arithmetic calculations.

foil. A transparency for overhead projection.

font. A particular style of typeface (for example, Gothic English). In GDDM, a font can exist as a programmed symbol set.

formatted document. A type of file containing text, images, and graphics.

FORTRAN. One of the programming languages supported by GDDM.

four-button cursor. A hand-held device, with cross-hair sight, used on the surface of a *tablet* to indicate position on a screen. Synonymous with *puck*.

frame. In GDDM-IMD, a synonym for *panel*.

full-screen alphanumeric operation. Full-screen processing operations on alphanumeric fields.

full-screen mode. A form of screen presentation in which the contents of an entire terminal screen can be displayed at once. Full-screen mode is often used for fill-the-blanks prompting, and is an alternative to line-by-line I/O.

full-screen processor. A host software component that, together with display terminal functions, supports display terminal input/output in full-screen mode.

G

GDDM. Graphical Data Display Manager. A series of IBM licensed programs, running in a host computer, that manage communications between application programs and display devices, printers, plotters, and scanners for graphics applications.

GDDM-GKS. GDDM Graphical Kernel System. A member of the GDDM family that runs under TSO and CMS and provides an alternative graphics programming interface to that of the GDDM base product. It is an implementation of the Graphical Kernel Standard, ISO 7942, of the International Organization for Standardization.

GDDM/graPHIGS. A member of the GDDM family used for creating hierarchical three-dimensional structures on the IBM 5080 Graphics System. It is based on the proposed ANSI standard for the Programmer's Hierarchical Interactive Graphics System (PHIGS).

GDDM Interactive Map Definition. GDDM-IMD. A member of the GDDM family of licensed programs. It enables users to create alphanumeric layouts at the terminal. The user defines the position of each field within the layout and may assign attributes, default data, and associated variable names to each field. The resultant map can be tested from within the utility.

GDDM-IVU. GDDM Interactive View Utility. A member of the GDDM family of licensed programs. It enables users to view, create, modify, store, and print images.

GDDM-OS/2. A licensed program that enables IBM PS/2 and other personal-computer systems with OS/2 installed to run GDDM application programs in the host computer.

GDDM-PCLK. A licensed program that enables IBM PS/2 and other personal computers with graphics-display adapters, and IBM 3270 terminal emulators to run GDDM application programs in the host computer.

GDDM-PGF. GDDM-Presentation Graphics Facility. A member of the GDDM family of licensed programs. It is concerned with business graphics, rather than general graphics.

GDDM storage. The portion of host computer main storage used by GDDM.

GDF. Graphics data format.

general graphics. The methods and techniques for converting data to or from graphics display in mathematical, scientific, or engineering applications; that is, in any application other than business graphics. See also *business graphics*.

generated mapgroup. The output produced when a source GDDM-IMD mapgroup is generated. It contains the information needed by GDDM at execution to position the mapped fields on the GDDM page.

| **GIF.** Graphics Interchange Format.

GKS. Graphical Kernel System. See *GDDM-GKS*.

GL. Graphical Language.

Graphical Data Display Manager. See *GDDM*.

graphics. A picture defined in terms of *graphics primitives* and *graphics attributes*.

graphics area. Part of a mapped field that is reserved for later insertion of graphics.

graphics attributes. In GDDM, color selection, color mix, line type, line width, graphics text attributes, marker symbol, and shading pattern definition.

graphics cursor. A physical indicator that can be moved (often with a joystick, mouse, or stylus) to any position on the screen.

graphics data format (GDF). A picture definition in an encoded order format used internally by GDDM and, optionally, providing the user with a lower-level programming interface than the GDDM API.

graphics data stream. The data stream that produces graphics on the screen, printer, or plotter.

graphics field. A rectangular area of a screen or printer page, used for graphics. Contrast with *alphanumeric field*.

graphics input queue. A queue associated with the graphics field onto which elements arrive from logical input devices. The program can remove elements from the queue by issuing a graphics read.

graphics primitive. A single item of drawn graphics, such as a line, arc, or graphics text string. See also *graphics segment*.

graphics read. A form of read that solicits graphics input or removes existing elements from the graphics input queue.

graphics segment. A group of graphics primitives (lines, arcs, and text) that have a common window and a common viewport and associated attributes. Graphics

segments allow a group of primitives to be subject to various operations. See also *graphics primitive*.

graphics text attributes. In GDDM, the symbol (character) set to be used, character box size, character angle, character mode, character shear angle, and character direction.

graPHIGS. See *GDDM/graPHIGS*.

gray-level. A digitally encoded shade of gray, normally (and always in GDDM) in the range 0 through 255. See also *gray-scale image*.

gray-scale image. An image in which the gradations between black and white are represented by discrete gray-levels. Contrast with *bi-level image* and *halftone image*.

green lightning. The name given to the flashing streaks on an IBM 3270 screen while a programmable symbol set is being loaded.

H

halftone image. A bi-level image in which intermediate shades of gray are simulated by patterns of adjacent black and white pixels. Contrast with *gray-scale image*.

Hangeul. A character set of symbols used in Korean ideographic alphabets.

hardware cell. Synonym for *character cell*.

hardware characters. Synonym for *hardware symbols*.

hardware symbols. The characters that are supplied with the device. The term is loosely used also for GDDM mode-1 symbols that are loaded into a PS store for subsequent display. Synonymous with *hardware characters*.

hexadecimal. Pertaining to a numbering system with base sixteen.

host. See *host computer*.

high performance alphanumerics. The creation of alphanumeric displays using field list data structures. Contrast with *procedural* and *mapped* alphanumerics.

host computer. The primary or controlling computer in a multiple-computer installation.

I

ICU. Interactive Chart Utility.

identity projection. In GDDM image processing, a projection that is transferred from source image to target image without any processing being performed on it.

image. Synonym for *digital image*.

image data stream. The internal form of the GDDM data in an image environment.

image field. A rectangular area of a screen or printer page, used for image. Contrast with *alphanumeric field* and *graphics field*.

Image Object Content Architecture (IOCA). An architected collection of constructs used to interchange and present images.

image symbol. A character or symbol defined as a dot pattern.

Image Symbol Editor (ISE). A GDDM-supplied interactive editor that enables users to create or modify their own image symbol sets (ISS).

image symbol set (ISS). A set of symbols each of which was created as a pattern of dots. Contrast with *vector symbol set (VSS)*.

IMD. See *GDDM Interactive Map definition*.

IMS/VS. Information Management System/Virtual Storage. A subsystem of MVS under which GDDM can be used.

include member. A collection of source statements stored as a library member for later inclusion in a compilation.

input queue. See *graphics input queue*.

integer. A whole number (for example, -2, 3, 457).

Intelligent Printer Data Stream (IPDS). A structured-field data stream for managing and controlling printer processes, allowing both data and controls to be sent to the printer. GDDM uses IPDS to communicate with the IBM 4224 printer.

Interactive Chart Utility (ICU). A GDDM-PGF menu-driven program that allows business charts to be created interactively by nonprogrammers.

interactive graphics. In GDDM, those graphics that can be moved or manipulated by a user at a terminal.

Interactive Map definition. A member of the GDDM family of licensed programs. It enables users to create alphanumeric layouts at the terminal. The operator defines the position of each field within the layout and may assign attributes, default data, and associated variable names to each field. The resultant map can be tested from within the utility.

interactive mode. A mode of application operation in which each entry receives a response from a system or program, as in an inquiry system or an airline reservation system. An interactive system can also be conversational, implying a continuous dialog between the user and the system.

interactive subsystem. (1) One or more terminals, printers, and any associated local controllers capable of operation in interactive mode. (2) One or more system programs or program products that enable user applications to operate in interactive mode; for example, CICS.

intercept. In a chart, a method of describing the position of one axis relative to another. For example, the x axis can be specified so that it intercepts (crosses) the y axis at the bottom, middle, or top of the plotting area of a chart.

inter-device copy. The ability to copy a page or the graphics field from the current primary device to another device. The target device is known as the alternate device.

IOCA. See *Image Object content Architecture*.

IPDS. See *Intelligent Printer Data Stream*.

ISE. Image Symbol Editor.

ISO. International Organization for Standardization.

ISPF. Interactive System Productivity Facility.

ISS. Image symbol set.

IVU. Image View Utility. See *GDDM-IVU*.

J

joystick. A lever that can pivot in all directions in a horizontal plane, used as a *locator* device.

K

Kanji. A character set of symbols used in Japanese ideographic alphabets.

Katakana. A character set of symbols used in one of the two common Japanese phonetic alphabets; Katakana is used primarily to write foreign words phonetically. See also *Kanji*.

key. In a legend, a symbol and an associated data group name. A key might, for example, indicate that the blue line on a graph represents "Predicted Profit." See also *legend*.

key symbol. A small part of a line (from a line graph) or an area (from a shaded chart) used in a legend to identify one of the various data groups.

L

Latin. Of or pertaining to the Western alphabet. In GDDM, a synonym for *single-byte character set*.

legend. A set of symbolic keys used to identify the data groups in a business chart.

line attributes. In GDDM, color, line type, and line width.

link pack area. An MVS term that describes an area of shared storage.

link-attached. Pertaining to devices that are connected to a controlling unit by a data link. Synonymous with *remote*. Contrast with *channel-attached*.

local. Synonym for *channel-attached*.

local character set identifier. A hexadecimal value stored with a GDDM symbol set, which can be used by symbol-set-loading means other than GDDM in the context of local copy on a printer.

locator. A logical input device used to indicate a position on the screen. Its physical form may be the alphanumeric cursor or a graphics cursor moved by a joystick.

logical input device. A concept that allows application programs to be written in a device-independent manner. The logical input devices to which the program refers may be subsequently associated with different physical parts of a terminal, depending on which device is used at run time.

LPA. Link pack area.

LTERM. In IMS/VS, logical terminal.

M

map. A predefined format of alphanumeric fields on a screen. Usually constructed outside of the application program.

map specification library (MSL). The data set in which maps are held in their source form.

mapgroup. A data item that contains a number of maps and information about the device on which those maps are to be used. All maps on a GDDM page must come from the same mapgroup.

mapped alphanumerics. The creation of alphanumeric displays using predefined maps. Contrast with *procedural alphanumerics* and *high performance alphanumerics*.

mapped field. An area of a page whose layout is defined by a map.

mapped graphics. Graphics placed in a graphics area within a mapped field.

mapped page. A GDDM page whose content is defined by maps in a mapgroup.

mapping. The use of a map to produce a panel from an output record, or an input record from a panel.

marker. In GDDM, a symbol centered on a point. Line graphs and polar charts can use markers to indicate the plotted points.

MDT. Modified data tag.

menu. A displayed list of logically grouped functions from which the user can make a selection. Sometimes called a menu panel.

menu-driven. Describes a program that is driven by user response to one or more displayed menus.

MFS. Message format service.

MICR. Magnetic ink character recognition.

mixed character string. A string containing a mixture of *Latin* (one-byte) and *Kanji* or *Hangeul* (two-byte) characters.

Mixed Object Document Content Architecture (MO:DCA). An architected, device-independent data stream for interchanging documents.

mode-1/-2/-3 characters. See *character mode*.

mountain shading. A method of shading surface charts where each component is shaded separately from the base line, instead of being shaded from the data line of the previous component.

mouse. A device that a user moves on a flat surface to position a pointer on a screen.

MSHP. Maintain System History Program. A software process for installing licensed programs on VSE systems.

MSL. Map specification library.

MVS. IBM Multiple Virtual Storage. A system under which GDDM can be used.

MVS/XA. Multiple Virtual Storage/Extended Architecture. A subsystem under which GDDM can be used.

N

name-list. A means of identifying which physical device is to be opened by a GDDM program. It can be used as a parameter of the DSOPEN call, or in a *nickname*.

National Language Support (NLS). A special feature that provides translations of the ICU panels and some of the GDDM messages into a variety of languages, including US English.

negate. In bi-level image data, setting zero bits to one and one bits to zero.

neutral color. White on a display, black on a printer. Contrast with *background color*.

nibblized data. The encoded data stream used between the GDDM DOS Support feature in the host and GDDM-PCLK on the workstation.

nickname. In GDDM, a means of referring to a device, the characteristics and identity of which have been already defined.

NLS. National Language Support.

nonqueriable printer. A printer about which GDDM cannot obtain any information.

NSS. Named saved system (VM/XA and VM/ESA).

null character. An empty character represented by X'00' in the EBCDIC code. Such a character does not occupy a screen position.

O

operator reply mode. In GDDM, the mode of interaction available to the operator (display terminal user) with respect to the modification (or not) of alphanumeric character attributes for an input field.

operator window. Part of the display screen's surface on which the GDDM output of an application program can be shown. An operator window is controlled by the end user; contrast with *partition*. A *task manager* may create a window for each application program it is running.

outbound structured field. An element in IBM 3270 data streams from host to terminal with formatting that allows variable-length and multiple-field data to be sequentially translated by the receiver into its component fields without the receiver having to examine every byte.

P

page. In GDDM, the main unit of output and input. All specified alphanumerics and graphics are added to the current page. An output statement always sends the current page to the device, and an input statement always receives the current page from the device.

page printer. A printer, such as the IBM 3820 or IBM 4250, to which the host computer sends data in the form of a succession of formatted pages. Such devices can print pictorial data and text, and can position all output to pixel accuracy. The pixel density and the general print quality both often suffice as camera-ready copy for publications. Also known as *composed-page printer*.

page segment. A picture file in a form that can be printed. It can only be printed if it is embedded in a primary document. Also known as a *PSEGo* file.

panel. A predefined display that defines the locations and characteristics of alphanumeric fields on a display terminal. When the panel offers the operator a selection of alternatives it may be called a menu panel. Synonymous with *frame*.

partition. Part of the display screen's surface on which a page, or part of a page, of GDDM output can be shown. Two or more partitions can be created, each displaying a page, or part of a page, of output. A partition is controlled by the GDDM application; contrast with *operator window*.

partition set. A grouping of partitions that are intended for simultaneous display on a screen.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

PCB. In GDDM, program communication block (IMS/VS).

PCLK. See *GDDM-PCLK*.

PDS. Partitioned data set (MVS).

pel. Picture element. See *pixel*.

PGF. Presentation Graphics Facility. A member of the GDDM family of licensed programs. It is concerned with business graphics, rather than general graphics.

PHIGS. Programmer's Hierarchical Interactive Graphics System.

pick. The action of the operator in selecting part of a graphics display by placing the graphics cursor over it.

pick aperture. A rectangular or square box that is moved across the screen by the graphics cursor. An item must lie at least partially within the pick aperture before it can be picked.

pick device. A logical input device that allows the application to determine which part of the picture was selected (or picked) by the operator.

picture interchange format (PIF) file. In graphics systems, the type of file, containing picture data, that can be transferred between GDDM and an IBM 3270-PC/G, /GX, or /AT workstation.

picture space. In GDDM, an area of specified aspect ratio that lies within the graphics field. It is centered on the graphics field and defines the part of the graphics field in which graphics will be drawn.

PIF. Picture interchange format.

pixel. The smallest area of a display screen capable of being addressed and switched between visible and invisible states. Synonymous with *display point*, *pel*, and *picture element*.

PL/I. One of the programming languages supported by GDDM.

plotter. An output device that uses pens to draw its output on paper or transparency foils.

pointings. Pairs of x-y coordinates produced by an operator defining positions on a screen with a locator device, such as a *mouse*.

polar chart. A form of business chart where the x axis is circular and the y axis is radial.

polyfillet. In GDDM, a curve based on a sequence of lines. It is tangential to the end points of the first and last lines, and tangential also to the midpoints of all other lines.

polyline. A sequence of adjoining lines.

popping. A method of ordering data whereby each item in a list or sequence takes the value of the previous item in the list or sequence, and is then removed from the list; when this happens, the list or sequence of data is said to be "popped."

ppi. Pixels per inch.

PQE. Printer queue element.

presentation graphics. Computer graphics products or systems, the functions of which are primarily concerned with graphics output presentation. For example, the display of business planning bar charts.

preview chart. A small version of the current chart that can be displayed on ICU menu panels.

primary device. In GDDM, the main destination device for the application program's output, usually a display terminal. The default primary device is the user console. See also *alternate device*.

primitive. See *graphics primitive*.

primitive attribute. A specifiable characteristic of a graphics primitive. See *graphics attributes* and *graphics text attributes*.

Print Services Facility (PSF). An IBM licensed program for processing documents destined for the IBM 3800 Model 3 page printer.

print utility. A subsystem-dependent utility that sends print files from various origins to a queued printer.

procedural alphanumerics. The creation of alphanumeric displays using the GDDM alphanumeric API. Contrast with *mapped alphanumerics* and *high performance alphanumerics*.

processing option. Describes how a device's I/O is to be processed. It is a device-family-dependent and subsystem-dependent option that is specified when the device is opened (initialized). An example is the choice between CMS attention-handling protocols.

procopt. Processing option.

profile. In GDDM, a file that contains information about how GDDM is to process requests for services to devices or other functions.

program library. (1) A collection of available computer programs and routines. (2) An organized collection of computer programs.

programmed symbols (PS). Dot patterns loaded by GDDM into the PS stores of an output device.

projection. In GDDM image processing, an application-defined function that specifies operations to be performed on data extracted from a source image. Consists of one or more *transforms*. See also *transform element*.

PS. Programmed symbols.

PS overflow. A condition where the graphics cannot be displayed in its entirety because the picture is too complex to be contained in the device's PS stores.

PSB. Program specification block (IMS).

PSEG. See *page segment*.

PSF. Print Services Facility.

PSP bucket. A database containing descriptions of faults found in programs. Used by Service personnel.

PS/2. Personal System/2.

puck. Synonym for *four-button cursor*.

PUT. Program update tape.

Q

quarto. A paper size, more common in the U.S. than in Europe. It measures 8.5 inches by 11.0 inches. Also known as A size. Compare with A4.

queued printer. A printer belonging to the subsystem under which GDDM runs, to which output is sent indirectly by means of the GDDM Print Utility program. In some subsystems, this may allow the printer to be shared between multiple users. Contrast with *system printer*.

R

raster device. A device with a display area consisting of dots. Contrast with *vector device*.

rastering. The transforming of graphics primitives into a dot pattern for line-by-line sequential use. In GDDM PS devices, this is done by transforming the primitives into a series of programmed symbols (PS).

glossary

real device. A GDDM device that is not being windowed by means of operator window functions. Contrast with *virtual device*.

reentrant. The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

remote. Synonym for *link-attached*.

reply mode. See *operator reply mode*.

resolution. In graphics and image processing, the number of pixels per unit of measure (inch or meter).

reverse clipping. Where one graphics primitive overlaps another, removing any parts of the underlying primitive that are overpainted by the overlying primitive.

reverse video. A form of alphanumeric highlighting for a character, field, or cursor, in which its color is exchanged with that of its background. For example, changing a red character on a black background to a black character on a red background.

REXX. Restructured Extended Executor Language. One of the programming languages supported by GDDM.

Roman. Relating to the *Latin* type style, with upright characters.

S

SBCS. Single-byte character set.

scanner. A device that produces a digital image from a document.

scissoring. Synonym for *clipping*.

scrolling. In computer graphics, moving a display image vertically or horizontally in a manner such that new data appears at one edge as existing data disappears at the opposite edge.

SCS. SNA character string.

segment. See *graphics segment*.

segment attributes. Attributes that apply to the segment as an entity, rather than to the individual primitives within the segment. For example, the visibility, transformability, or detectability of a segment.

segment library. The portion of auxiliary storage where segment definitions are held. These definitions are GDDM objects in graphics data format (GDF) and are managed by GDDM API calls. GDDM handles the file accesses to and from auxiliary storage.

segment priority. The order in which segments are drawn; also the order in which they are detected.

segment transform. The means to rotate, scale, and reposition segments without re-creating them.

selector adjunct. A subfield of an application data structure that qualifies a data field.

shear. The action of tilting graphics text so that each character leans to the left or right while retaining a horizontal baseline.

single-byte character set (SBCS). A set of characters in which each character occupies one byte position in internal storage and in display buffers. Used for example, in most non-Oriental symbols. Contrast with *double-byte character set*.

SMP/E. System Modification Program/Extended. A software process for installing licensed programs on MVS systems.

SNA. System Network Architecture.

source image. An image that is the data input to image processing or transfer.

spill file. A means of reducing storage requirements at the cost of processing time, when creating high-resolution output files for page printers, for example.

stand-alone (mode). Operation that is independent of another device, program, or system.

standard default. The value of a graphics attribute chosen by GDDM when no value is explicitly specified by the user. The standard default cannot be altered by the user, although it may be overridden by the user.

string device. A logical input device that enables an application program to process character data entered by the terminal operator.

stroke device. A logical input device that enables an application program to process a sequence of x,y coordinate data entered by the terminal operator.

stylus. A pen-like pointer used on the surface of a tablet to indicate position on a screen.

surface chart. A chart similar to a line graph, except that no markers appear and the areas between successive lines are shaded.

swathe. A horizontal slice of printer output, forming part of a complete picture. Page printer images are often constructed in swathes to reduce the amount of storage required.

symbol. Synonymous with *character*. For example, the following terms all have the same meaning: vector symbols, vector characters, vector text.

symbol cell. Synonym for *character cell*.

symbol matrix. Synonym for *dot matrix*.

symbol set. A collection of symbols, usually but not necessarily forming a font. GDDM applications may use the hardware device's own symbol set. Alternatively, they can use image or vector symbol sets that the user has created.

symbol set identifier. In GDDM, an integer (or the equivalent EBCDIC character) by which the programmer refers to a loaded symbol set.

system printer. A printer belonging to the subsystem under which GDDM runs, to which output is sent indirectly by use of system spooling facilities. Contrast with *queued printer*.

T

tablet. (1) A locator device with a flat surface and a mechanism that converts indicated positions on the surface into coordinate data. (2) The IBM 5083 Tablet Model 2, which, with a four-button cursor or stylus, allows positions on the screen to be addressed and the graphics cursor to be moved without use of the keyboard.

tag. In interactive graphics, an identifier associated with one or more primitives that is returned to the program if such primitives are subsequently picked.

target image. An image that is the destination of processed or transferred data.

target position. In the GDDM Vector Symbol Editor, the grid coordinates of a point on the editing grid to which a vector is to be drawn.

task manager. A program that supervises the concurrent running of other programs.

temporary graphics. Graphics created outside a segment.

terminal. A device, usually equipped with a keyboard and a display unit, capable of sending and receiving information over a link. See also *display terminal*.

terminal emulator. A program that enables a device such as a personal computer system to enter and receive data from a host computer system as if it were a particular type of attached terminal.

test symbol. In the GDDM Image and Vector Symbol Editors, an area on the Symbol Edit panel in which the currently chosen symbol is displayed.

text. Characters or symbols sent to the device. GDDM provides alphanumeric text and graphics text.

text attributes. See *graphics text attributes*.

tilted pie chart. A pie chart drawn in three dimensions, which has been tilted away from full face to reveal its three-dimensional properties.

trailing attribute byte. The screen position following an alphanumeric field. This attribute byte can specify, for example, that the cursor should auto-skip to the next field when the current field is filled.

transfer operation. In GDDM image processing, an operation in which a projection is applied to a source image, and the result placed in a target image. The source and target images can be device or application images in any combination, or one or other of them (but not both) can be image data within the application program.

transform. (1) The action of modifying a picture for display; for example, by scaling, rotating, or displacing. (2) The object that performs or defines such a modification; also referred to as a *transformation*. (3) In GDDM image processing, a definition of three aspects of the data manipulation to be done by a projection:

1. A transform element or sequence of transform elements
2. A resolution conversion or scaling algorithm
3. A location within the target image for the result

Only the third item is mandatory.

See also *projection* and *transform element*.

transform element. In GDDM image processing, a specific function in a transform, which can be one of the following: define sub-image, scale, orient, reflect, negate, define place in target image.

A given transform element can be used only once in a *transform*.

transformable. A segment must be defined as transformable if it will subsequently be moved, scaled, or rotated.

transparency. (1) A document on transparent material suitable for overhead projection. (2) An alphanumeric attribute that allows underlying graphics or image to show.

TSO. Time Sharing Option. A subsystem of MVS under which GDDM can be used.

TWA. Transaction work area.

U

UDS. User default specification.

UDSL. A list of user default specifications (UDSs).

unformatted data. In GDDM image processing, compressed or uncompressed binary image data that has no headers, trailers, or embedded control fields other than any defined by the compression algorithm, if applicable. The data is in row major order, beginning with the top left of the picture.

User Control. A GDDM function that enables the terminal or workstation to perform some functions without the need for application programming. The actions include: moving and zooming graphics; manipulating windows; printing, plotting, and saving pictures.

user default specification (UDS). The means of changing a GDDM external default value. The external default values that a UDS can change are those of the GDDM or subsystem environment, GDDM user exits, and device definitions.

user exit. A point in GDDM execution where a user routine will gain control if such has been requested.

V

variable cell size. In most devices, the hardware cell size is fixed, but the IBM 3290 Information Panel has a cell size that can be varied. This, in turn, causes the number of rows or columns on the device to alter.

vector. (1) In computer graphics, a directed line segment. (2) In the GDDM-PGF Vector Symbol Editor, a straight line between two points.

vector device. A device capable of displaying lines and curves directly. Contrast with *raster device*.

vector symbol. A character or shape composed of a series of lines or curves.

Vector Symbol Editor. A program supplied with GDDM-PGF, the function of which is to create and edit vector symbol sets (VSS).

vector symbol set (VSS). A set of symbols, each of which was originally created as a series of lines and curves.

Venetian blind effect. The name given to the appearance of bars across shaded patterns on an IBM 3270-PC when GDDM tries to match the image symbol sets.

Venn diagram. A form of business chart in which, in GDDM, two or more populations and their intersection are represented by overlapping circles.

viewport. A subdivision of the picture space, most often used when two separate pictures are to be displayed together.

virtual device. A GDDM device that is being windowed by use of operator window functions. Contrast with *real device*.

virtual screen. The presentation space viewed through an *operator window*.

VM/ESA. IBM Virtual Machine Enterprise Systems Architecture.

VM/SP CMS. IBM Virtual Machine/System Product Conversational Monitor System; a system under which GDDM can be used.

VMXA. IBM Virtual Machine Extended Architecture; a system under which GDDM can be used.

VSE. Virtual storage extended; an operating system consisting of VSE/Advanced Functions and other IBM programs.

Note: In GDDM, the abbreviation VSE has sometimes been used to refer to the Vector Symbol Editor, but to avoid confusion, this usage is deprecated.

VSS. Vector symbol set.

W

Ward. One of the 190 matrices used to contain the symbols of a double-byte character set. The value in the first byte of each double-byte character code refers to the ward in which the character is contained. The value in the second byte denotes the character's position in the matrix.

window. In GDDM, the term window has three distinct meanings:

1. The "graphics window" is the coordinate space used for defining the primitives that make up a graphics picture. By default, both x and y coordinates run from 0 through 100. The graphics window can be regarded as a set of coordinates that are overlaid on the viewport.
2. An "operator window" is an independent rectangular subdivision of the screen. Several can exist at the same time, and each can receive output from, and send input to, either a separate GDDM program or a separate function of a single GDDM program.

3. The “page window” defines which part should be displayed of a page that is deeper or wider than its partition.

workstation. A display screen together with attachments such as a local copy device or a tablet.

world coordinates. The user application-oriented coordinates used for drawing graphics. See also *window*.

wrap-around field. An alphanumeric field that extends to the right-hand edge of the page and continues at the start of the next row.

WTP. Write-to-programmer.

Bibliography

The books of the GDDM library are listed here.

Latest GDDM information

- | For up-to-date information on GDDM products, check our Home Page on the Internet at the following URL:
- | <http://www.hursley.ibm.com/gddm/>
- | You might also like to look at the IBM Software Home Page at:
- | <http://www.software.ibm.com>

GDDM publications

- | | |
|-----------|--|
| GDDM Base | <i>GDDM Base Application Programming Guide</i> , SC33-0867
<i>GDDM Base Application Programming Reference</i> , SC33-0868
<i>GDDM Diagnosis</i> , SC33-0870
<i>GDDM General Information</i> , GC33-0866
<i>GDDM/MVS Program Directory</i> , GC33-1801
<i>GDDM/VM Program Directory</i> , GC33-1802
<i>GDDM/VSE Program Directory</i> , GC33-1803
<i>GDDM Messages</i> , SC33-0869
<i>GDDM Series Licensed Program Specifications</i> , GC33-0876
<i>GDDM System Customization and Administration</i> , SC33-0871
<i>GDDM User's Guide</i> , SC33-0875
<i>GDDM Using the Image Symbol Editor</i> , SC33-0920 |
| GDDM-GKS | <i>GDDM-GKS Programming Guide and Reference</i> , SC33-0334 |
| GDDM-IMD | <i>GDDM Interactive Map Definition</i> , SC33-0338 |
| GDDM-IVU | <i>GDDM Image View Utility</i> , SC33-0479 |
| GDDM-PGF | <i>GDDM-PGF Application Programming Guide</i> , SC33-0913
<i>GDDM-PGF Programming Reference</i> , SC33-0333
<i>GDDM-PGF Interactive Chart Utility</i> , SC33-0328
<i>GDDM-PGF Vector Symbol Editor</i> , SC33-0330
<i>GDDM-PGF OPS User's Guide</i> , SC33-1776 |
- | GDDM/MVS is an element of OS/390. GDDM-REXX/MVS and GDDM-PGF are optional features of OS/390. For a
 - | complete list of the publications associated with OS/390, see the *OS/390 Information Roadmap*, GC28-1727.

Books from related libraries

In addition to the GDDM library, you may need to refer to some of these books:

APL2	<i>APL2 Installation and Customization under CMS</i> , SH21-1062 <i>APL2 Installation and Customization under TSO</i> , SH21-1055 <i>APL2 Messages and Codes</i> , SH21-1059 <i>APL2 Diagnosis</i> , LY27-9601
CBIPO	<i>SMP/E CBIPO Dialogs User's Guide</i> , SC23-0538
CBPDO	<i>Understanding and Using CBPDO</i> , GG66-0290
CICS	<i>CICS/ESA Installation Guide</i> , SC33-0663 <i>System Definition Guide</i> , SC33-0664 <i>Operations Guide</i> , SC33-0668 <i>Resource Definition (Online)</i> , SC33-0666 <i>Resource Definition (Macro)</i> , SC33-0667 <i>CICS/VSE System Definition and Operations Guide</i> , SC33-0716 <i>Resource Definition (Macro)</i> , SC33-0667 <i>Resource Definition (Online)</i> , SC33-0666
Composed Document Printing Facility	<i>Installation and Operations</i> , SC33-6135
DOS	<i>DOS 5.00 User's Guide and Reference</i> , Z84F-9779
GDDM/graPHIGS	<i>Installing GDDM/graPHIGS</i> , SC33-8101 <i>Licensed Program Specifications</i> , GH23-0001 <i>Introducing graPHIGS</i> , SC33-8190 <i>Understanding graPHIGS</i> , SC33-8102 <i>Writing Applications with graPHIGS</i> , SC33-8103 <i>Programmer's Reference for graPHIGS</i> , SC33-8104 <i>Messages and Error Codes for graPHIGS</i> , SC33-8105 <i>Programmer's Pocket Reference for graPHIGS</i> , SC33-8107 <i>Problem Diagnosis for graPHIGS</i> , SC33-8108
IMS	<i>IMS/VS Installation Guide</i> , SH20-9081 <i>IMS/VS Messages and Codes Reference Manual</i> , SH20-9030 <i>IMS/VS System Programming Reference Manual</i> , SH20-9027 <i>IMS/VS Utilities Reference Manual</i> , SH20-9029
IPDS	<i>Intelligent Printer Data Stream Reference</i> , GA34-2082 <i>IBM 3812 and 3816 Page Printers: IPDS Handbook</i> , GA34-2082
ISPF	<i>ISPF User's Guide</i> , SC34-4484
MSHP	<i>VSE Advanced Function: MSHP Reference</i> , SC33-6199 <i>VSE/ESA: System Control Statements</i> , SC33-6513
Networking	<i>Network Program Products Samples: VM SNA</i> , SC30-3309
Operating System/2	Use the following generic titles that apply to the level of OS/2 you are using: <i>OS/2 Information and Planning Guide</i> <i>OS/2 System Administrator's Guide for Communications</i> <i>OS/2 Getting Started</i> <i>OS/2 User's Guide</i> <i>OS/2 EHLLAPI Programming Reference</i>
Personal Services/CICS	<i>Installing and Customizing Personal Services/CICS (MVS)</i> , SC30-3294 <i>Installing and Customizing Personal Services/CICS (VSE)</i> , SC30-3295
Print Services Facility	<i>System Programmer's Guide for MVS</i> , SH35-0091

	<i>System Programmer's Guide for VM</i> , S544-3511
	<i>System Programmer's Guide for VSE</i> , S544-3103
SMP/E	<i>SMP/E Terminal User's Guide</i> , SC28-1109 <i>SMP/E User's Guide</i> , SC28-1302 <i>SMP/E Reference</i> , SC28-1107
TSO	<i>VS APL for TSO: Terminal User's Guide</i> , SH20-9180 <i>APL2 Installation and Customization under TSO</i> , SH21-1055 <i>JES/328X Print Facility — Version 2 Release 2 Program Offering; Program Description and Operator's Manual</i> , SH20-7174 <i>MVS/XA TSO Guide to Writing a Terminal Monitor Program or a Command Processor</i> , GC28-1295 <i>TSO/E Version 2, Procedures Language, MVS/REXX Reference</i> , SC28-1883 <i>TSO/E Command Reference</i> , SC28-1881
VMSES/E	<i>VM/ESA: VMSES/E Introduction and Reference</i> , SC24-5444
VM/ESA	<i>VM/ESA CP Planning and Administration</i> , SC24-5521 <i>VM/ESA CMS Planning and Administration</i> , SC24-5445 <i>VM/ESA Installation Guide</i> , SC24-5526 <i>VM/ESA Procedures Language VM/REXX Reference</i> , SC24-5466 <i>VM/ESA CP Command and Utilities Reference</i> , SC24-5519 <i>VM/ESA CMS Command Reference</i> , SC24-5461
VSAM	<i>Using VSE/VSAM Commands and Macros</i> , SC24-5144 <i>VSE/VSAM Programmer's Reference</i> , SC24-5145 <i>VSE/VSAM Messages and Codes</i> , SC24-5146 <i>MVS/XA VSAM Catalog Administration Access Method Services Reference</i> , GC26-4136
VSE/ESA	<i>VSE/ESA Planning</i> , SC33-6503 <i>VSE/ESA Installation</i> , SC33-6504 <i>VSE/ESA System Control Statements</i> , SC33-6513
3117 scanner	<i>IBM 3117 Scanner and IBM 3117 PC Adapter Guide to Operations</i> , GA18-2477 <i>IBM 3117 Scanner and Extension Unit Guide to Operations</i> , GA18-2478 <i>IBM 3117 Scanner Hardware Maintenance and Service</i> , SY18-2159 <i>IBM 3117 Scanner Technical Reference</i> , SC18-2105
3118 scanner	<i>Scanner Guide to Operations</i> , GA18-2475 <i>High Speed Adapter Guide to Operations</i> , GA18-2476 <i>IBM 3118 Scanner Hardware Maintenance and Service</i> , SY18-2158 <i>High Speed Adapter Hardware Maintenance and Service</i> , SY18-2167 <i>Scanner Technical Reference</i> , SC18-2104 <i>High Speed Adapter Technical Reference</i> , SC18-2117
3174 establishment controller	<i>Data Stream Programmer's Reference</i> , GA23-0059 <i>Functional Description</i> , GA23-0218 <i>Terminal User's Reference for Expanded Functions</i> , GA23-0332 <i>Planning Guide Configuration Support B Release 2</i> , GA27-3862 <i>Customer Problem Determination</i> , GA23-0217 <i>Status Codes</i> , GA27-3832
3179-G, 3192-G	<i>3179-G and 3192-G Color Graphics Display Station Description</i> , GA18-2589
3193 display station	<i>Description</i> , GA18-2364 <i>Setup Instructions</i> , GA18-2366 <i>Operator's Guide</i> , GA18-2365 <i>Problem Solving Quick Check Guide</i> , GA18-2443 <i>Problem Solving Guide</i> , GA18-2444
3270-family devices	<i>3270 Information Display System Configurator</i> , GA27-2849 <i>3270 Information Display System Data Stream Programmer's Reference</i> , GA23-0059 <i>8775 Display Terminal: Component Description</i> , GA33-3044

bibliography

3270-PC/G and 3270-PC/GX workstations	<i>IBM 3270 Information Display System: Color and Programmed Symbols</i> , GA33-3056 <i>Introducing the IBM 3270 Personal Computer/G and /GX Ranges of Workstations</i> , GA33-3157 <i>3270-PC/G Personal Computer/G and /GX Ranges of Workstations; Planning Guide</i> , GA33-3158 <i>3270-PC/G Guide to Operations</i> , SA33-3155 <i>3270-PC/GX Guide to Operations</i> , SA33-3156
3274 control unit	<i>3274 Control Unit Description and Programmer's Guide</i> , GA23-0061 <i>3274 Control Unit Planning, Setup and Customization Guide</i> , GA27-2827
3472-G display	<i>3472-G User's Guide</i> , GA18-7026
3812 printer	<i>IPDS NDS Attachment Feature Installation and Programming Instructions</i> , S544-3101 <i>Guide to Operations</i> , S544-3267
3816 page printer	<i>IBM 3816 Page Printer Operating Instructions</i> , GA34-2075
3820 page printer	<i>IBM 3820 Page Printer Operator's Guide</i> , S544-3080 <i>IBM 3820 Page Printer Reference Manual</i> , S544-3175
3825 page printer	<i>3825 Page Printer Operator's Guide</i> , G544-3481 <i>3825 Page Printer Product Description</i> , G544-3482
3827 page printer	<i>3827 Page Printer Operator's Guide</i> , G544-3189 <i>3827 Page Printer Product Description</i> , G544-3194
3828 advanced function MICR printer	<i>IBM 3828 Advanced Function MICR Printer Operator's Guide</i> , S544-3360 <i>IBM 3828 Advanced Function MICR Printer Product Description</i> , S544-3361
3835 page printer	<i>3835 Page Printer Product Description</i> , G544-3498 <i>3835 Page Printer Operator's Guide</i> , G544-3208
3900 advanced function printer	<i>IBM 3900 Advanced Function Printer Product Description</i> , GA32-0135 <i>IBM 3900 Advanced Function Printer Operator's Guide</i> , GA37-0210
4028 printer	<i>IBM LaserPrinter 4028 Introduction and Planning Guide</i> , S544-4258 <i>IBM LaserPrinter 4028 IPDS Handbook</i> , S544-4260 <i>3270 Programming Guide and Reference Manual for the IBM LaserPrinter 4028 Model NS1</i> , S544-4262 <i>IBM LaserPrinter 4028 Model NS1 Guide to Operations</i> , S544-4263
4224 printer	<i>Printer Product and Programming Description Manual</i> , GC31-2551 <i>Operating Instructions</i> , GC31-2546 <i>Guide to Operations</i> , GC31-3621
4230 printer	<i>4230 Models 102/202 Printer Product and Programming Description</i> , GC40-1701 <i>4230 Models 102/202 User's Guide</i> , SA40-0564 <i>4230 Models 102/202 Operator's Panel Instructions</i> , SA40-0565
4234 printer	<i>4234-11,12,13 Product Description and Programming Manual</i> , GC31-3879 <i>4234-11 Operating Instructions</i> , GC31-3736
4250 printer	<i>Operator's Guide</i> , GA33-1551
5550 multistation	<i>5550 Japanese 3270-PC User's Guide</i> , N:SC18-2059 <i>How To Use 5550 Japanese 3270-PC</i> , N:SC18-2060 <i>5550 Japanese 3270-PC/G User's Guide</i> , N:SC18-2071 <i>How To Use 5550 Japanese 3270-PC/G</i> , N:SC18-2072 <i>5550 Small Cluster User's Guide</i> , N:SC18-2092 <i>How To Use 5550 Small Cluster</i> , N:SC18-2091 <i>5550 Small Cluster/Graphics User's Guide</i> , N:SC18-2107 <i>How To Use 5550 Small Cluster/Graphics</i> , N:SC18-2108 <i>5550 3270 Kanji Emulation Description</i> , N:SC18-2020 <i>5550 3270 Kanji Emulation Operator's Guide</i> , N:SC18-2021

Index

Numerics

3174 controller diagnosis 11
 3179-G graphics diagnosis 6
 3192-G graphics diagnosis 6
 3193 diagnosis 9
 3268
 printer diagnosis 9
 3270 EDCB verification 7
 3274 11
 controller diagnosis 10
 3278 graphics diagnosis 6
 3279 graphics diagnosis 6
 3287
 printer diagnosis 8
 3472-M graphics diagnosis 6
 3812 13
 4224 and 4234 printer diagnosis 7
 5080IO function in TRCESTR statements 38
 example 55

A

AAB (application anchor block) 144
 abend
 codes, list of 173
 dump, specification in trace qualifier 72
 reporting 139
 abend codes 173
 1201 13
 2201 13
 G201 13
 ABEND function in TRCESTR statements 31
 abend problems 13
 ACA (AIC control area) 144
 action functions in TRCESTR statements 30
 address, storage 86
 ADM0275 message, graphics (image) cannot be shown 14
 ADMADFC defaults module (CICS)
 trace control word in 70
 TRCESTR statements in source of 69
 ADMADFD defaults module (VSE)
 trace control word in 70
 TRCESTR statements in source of 69
 ADMADFI 24
 ADMADFI defaults module (IMS)
 trace control word in 70
 TRCESTR statements in source of 69
 ADMADFT defaults module (TSO)
 trace control word in 70
 TRCESTR statements in source of 69

ADMADFV defaults module (VM/CMS)
 trace control word in 70
 ADMADFV defaults module (VM)
 TRCESTR statements in source of 69
 ADMMDFT
 DEFAULT validity 27
 macro 69
 TRACE default 69
 TRACE value 69
 TRCESTR statements 69
 ADMTAAB (application anchor block) 144
 ADMTACA (AIC control area) 144
 ADMTCQYV (VM display communication block) 148
 ADMTDFT (general defaults table) 151
 ADMTDSA (dynamic save area) 156
 ADMTDSAO (dynamic save area overflow stack) 158
 ADMTDSAS (dynamic save area stack) 158
 ADMTIFCB (interface control block) 159
 ADMTIFCE (error control feedback) 163
 ADMTNICK (nicknames) 164
 ADMTTCA (terminal services interface control area) 165
 ADMTTDB (terminal descriptor block) 166
 ADMTTRB (terminal request block) 168
 ADMUTRCE 26
 advice, where to look for more 3
 ALL function in TRCESTR statements 38
 APAR 135
 including GDDM-IMD record validation messages 85
 number 142
 severity code 142
 submitting 142
 APAR (Authorized Program Analysis Report) 135
 application anchor block (ADMTAAB) 144
 application interface control area (ADMTACA) 144
 application programs
 problems with 3
 tracing in 25
 ASF
 IBM 4224 8
 authorized program analysis report
 See APAR

B

bibliography 229
 black lines on 3812 output 13
 blocks, control, locating 72

C

CDPDS function in TRCESTR statements 31
 CGM statement TRCESTR statements
 export 64
 import 62
 CGMREP function in TRCESTR statements 32
 checking hardware characteristics 6
 diagnosis on 3174 controller 11
 diagnosis on 3193 9
 diagnosis on 3268 printer 9
 diagnosis on 3274 controller 10
 diagnosis on 3287 printer 8
 EDCB verification 7
 general printer diagnosis 7
 graphics diagnosis on 3179-G and 3192-G 6
 graphics diagnosis on 3278 and 3279 6
 graphics diagnosis on 3472-M 6
 non-IBM devices 12
 CICS destination control table (DCT) 24
 CICS, tracing under 23
 CICSTRCE 27
 CLEAR function in TRCESTR statements 30
 CMSTRCE 27
 codes, sense 2
 coding TRCESTR statements 27
 commands
 GQFLEVEL 108
 pcl /s 98
 pclkev 98
 SYSLEVEL 107
 common installation errors 12
 component function in TRCESTR statements 39
 control block and stack, trace output,
 GDDM-PCLK 103
 control blocks 143
 ADMTAAB (application anchor block) 144
 ADMTDFT (general defaults table) 151
 ADMTDSA (dynamic save area) 156
 ADMTDSAO (dynamic save area overflow
 stack) 158
 ADMTDSAS (dynamic save area stack) 158
 ADMTIFCB (interface control block) 159
 ADMTIFCE (interface error feedback) 163
 ADMTNICK (nicknames) 164
 ADMTTCA (terminal services interface control
 area) 165
 ADMTTDB (terminal descriptor block) 166
 ADMTTRB (terminal request block) 168
 application interface control area (ADMTACA) 144
 locating from current module 73
 VM display communication block
 (ADMTQCQYV) 148
 control function in TRCESTR statements 42
 controller diagnosis
 3174 11

controller diagnosis (*continued*)
 3274 10
 controllers
 3174 11
 3274 10
 conventions xvii
 COUNT function in TRCESTR statements 39
 CPNIN trace record 43
 in in-storage trace table 76
 CPNOUT trace record 43
 in in-storage trace table 76
 cross-reference, message-to-module 185

D

data areas 143
 ADMTAAB (application anchor block) 144
 ADMTDFT (general defaults table) 151
 ADMTDSA (dynamic save area) 156
 ADMTDSAO (dynamic save area overflow
 stack) 158
 ADMTDSAS (dynamic save area stack) 158
 ADMTIFCB (interface control block) 159
 ADMTIFCE (interface error feedback) 163
 ADMTNICK (nicknames) 164
 ADMTTCA (terminal services interface control
 area) 165
 ADMTTDB (terminal descriptor block) 166
 ADMTTRB (terminal request block) 168
 application interface control area (ADMTACA) 144
 VM display communication block
 (ADMTQCQYV) 148
 DCB (device control block) 7
 DEBUG command (GDDM-IMD) 85
 DEFAULT
 See ADMMDFT
 device checks 2
 PROG752 19
 program check 19
 device control block (DCB) 7
 device program check, reporting 138
 device reporting 141
 DFS0089 message 15
 DFS2078 message 15
 DFS971I message 15
 DFT (general defaults table) 151
 diagnosis
 3193 9
 3268 printer 9
 3287 printer 8
 graphics on 3179-G and 3192-G 6
 graphics on 3278 and 3279 6
 graphics on 3472-M 6
 initial 1
 locating control blocks (including in-storage trace
 table) 72

diagnosis (*continued*)
 non-IBM devices 12
 printer, general 7
 procedures 1
 diagnostic facility, GDDM-IMD 85
 diagnostic support
 GDDM-OS/2 Link 109
 GDDM-PCLK 98
 display communication block (ADMTCQYV) 148
 DISPLAY DUMP STATUS MVS command 81
 DISPLAY DUMP TITLE MVS command 81
 display dump title under MVS 81
 displaying main storage 86
 DMKID546I message 16
 documentation problem, reporting 140
 DSA (dynamic save area) 156
 DSAO (dynamic save area overflow stack) 158
 DSAS (dynamic save area stack) 158
 DSOPEN function in TRCESTR statements 32
 DUMP MVS command 81
 dump title under MVS 81
 DUMPDS MVS command 81
 Dynamic Dump under MVS 80
 dynamic save area (ADMTDSA) 156
 dynamic save area overflow stack (ADMTDSAO) 158
 dynamic save area stack (ADMTDSAS) 158

E

EDCB verification 7
 ENTRY function in TRCESTR statements 40
 environment, GDDM-OS/2 Link 113
 ERFIN trace record 43, 44
 ERFOUT trace record 43, 44
 error
 diagnosis 1
 installation 12
 messages 2
 reporting 135
 user abends 13
 error data in log messages, how to interpret 121
 error log entries, communications manager 120
 error tracing
 GDDM-OS/2 Link 109
 GDDM-PCLK 98
 errors
 unexpected message encountered 14
 ESEUDS, GDDM call for tracing 69, 70
 ESSUDS, GDDM call for tracing 69, 70
 event functions in TRCESTR statements 38
 EXIT function in TRCESTR statements 40
 explanation, trace 43
 extended edition common services, GDDM-OS/2 Link
 how to interpret error data in log messages 121
 external defaults
 TRCESTR 22

F

failure, program 1
 fix for GDDM-PCLK supplied by IBM 135
 floating-point registers (FR), in TRCESTR
 statements 29
 example of trace output 52
 FREGS function 33
 FLOW function in TRCESTR statements 32
 FORCE function in TRCESTR statements 30
 format of in-storage trace table 75
 format of trace records 42
 FR
 See floating-point registers (FR)
 frame, diagnostic (GDDM-IMD) 85
 FREGS function in TRCESTR statements 33
 example 52
 FSTRCE statement 70
 abend dump output 72
 FSTRCE, GDDM call for tracing 70
 FULLIO function in TRCESTR statements 33
 example 49
 FULLTCA function in TRCESTR statements 33

G

GDDM
 abend, reporting 139
 documentation problem, reporting 140
 installation problems 5
 internal problems 5
 message, reporting 138
 print utilities, problems with 4
 tracing 21
 tracing from GDDM-PCLK 98
 tracing multiple instances 25, 57
 utility programs, problems with 3
 GDDM Interactive Map Definition
 diagnostic facility 85
 main storage display 86
 GDDM Internet home page 229
 GDDM-IMD
 See *also* GDDM Interactive Map Definition
 DEBUG command 85
 diagnostic frame 85
 GDDM-IVU 4
 GDDM-OS/2 Link
 diagnosis 107
 environment 113
 extended edition common services,
 GDDM-OS/2 Link 121
 how to interpret error data in log messages 121
 finding level of link code 108
 OS/2 Communication Manager trace
 (auto-start) 118
 OS/2 Communication Manager trace (normal
 operation) 117

index

GDDM-OS/2 Link (*continued*)
 Session n 115
 trace points windows 112
 trace support 109
 tracing through modules 116
 update window 112
 using communications manager error log entries 120
 using for error log entries 120
 window procedure 114
GDDM-PCLK
 commands 98
 control block and stack 103
 diagnosis 95
 Main Panel 98
 Main Panel with service functions 98
 Service Functions panel 99
 trace control keys 101
 trace message data file 103
 trace modules 101
 trace options 100
 trace output 101
 trace output, XHGI component 102
 trace points (GDDM-PCLK) 99
 trace support 98
 transmission buffer 102
GDDM-REXX
 diagnosis 91
GDDM-REXX under MVS
 diagnosing 92
GDDM-REXX under VM
 diagnosing 93
GDDM/graPHIGS 37, 38
general defaults table (ADMTDFT) 151
general-purpose registers (GR), in TRCESTR statements 29
 example of trace output 52
 GREGS function 33
GQFLEVEL command 108
GR
 See general-purpose registers (GR)
Graphics Control Program 4, 12
 errors 20
graphics diagnosis
 3179-G and 3192-G 6
 3278 and 3279 6
 3472-M 6
GREGS function in TRCESTR statements 33
 example 52

H

hardware characteristics, checking 6
 diagnosis on 3174 controller 11
 diagnosis on 3193 9
 diagnosis on 3268 printer 9

hardware characteristics, checking (*continued*)
 diagnosis on 3274 controller 10
 diagnosis on 3287 printer 8
 EDCB verification 7
 general printer diagnosis 7
 graphics diagnosis on 3179-G and 3192-G 6
 graphics diagnosis on 3278 and 3279 6
 graphics diagnosis on 3472-M 6
 non-IBM devices 12
hardware problems 4
HCPMHT2150I message 17
home page for GDDM 229
how to interpret error data in log messages,
 GDDM-OS/2 Link 121
HRIG function in TRCESTR statements 33

I

IBM 4224
 ASF 8
IBM Support Center 135
 contacting 142
 information requirements for GDDM-OS/2 Link 108
 information requirements for GDDM-PCLK 98
 reporting problems to 135
IEA911E message 82
IEE094D message 81
IFCB (interface control block) 159, 173
IFCE (interface control error feedback) 163
IMD
 See GDDM Interactive Map Definition
IMS, tracing under 24
IMSTRCE 27
in-storage trace tables
 locating 72, 74
 meaning of records 76
 record format 75
incorrect output, reporting 138
installation
 common errors and pitfalls 12
 of GDDM, problems 5
Interactive Map Definition
 See GDDM Interactive Map Definition
interface control block (ADMTIFCB) 159
interface control block (IFCB) 173
interface control error feedback (ADMTIFCE) 163
Internet home page for GDDM 229
IST211I message 17

L

level function in TRCESTR statements 40
level of code installed at personal computer system 108
 level of GDDM-OS/2 Link code 108
 level of GDDM-PCLK code 98

level of code installed at personal computer system

(*continued*)

level of OS/2 code 107

LIMIT function in TRCESTR statements 40

line time-outs 18

LIST function in TRCESTR statements 34

locating

control blocks 72

control blocks from current module 73

current module 73

in-storage trace table 74

logoff 14, 16, 17, 18

loop, reporting 139

M

machine check 207 18

Main Panel of GDDM-PCLK

with service functions 98

main storage, display 86

map specification library (MSL) trace 87

message-to-module cross-reference 185

messages

ADM0275 14

beginning ADM 14

beginning AEM 15

beginning ERX 16

beginning GQD 16

beginning GQF 17

DFS0089 15

DFS2078 15

DFS971I 15

DMKID546I 16

error 2

HCPMHT2150I 17

IST211I 17

reporting 138

missing interrupt condition 14, 16, 17, 18

missing output, reporting 138

MODIN trace record 43, 44

in in-storage trace table 77

MODOUT trace record 43, 44

in in-storage trace table 77

modules

current, locating 73

trace output, GDDM-PCLK 101

MSGOUT trace record 43

MSL trace

access trace 88

introduction and use 87

record trace 88

record validation 89

multiple instances of GDDM, tracing in 25

example 57

N

NAME function in TRCESTR statements 41

NCP SLOWDOWN condition

CICS under MVS 18

IMS 18

NICK (nickname list entry format) 164

nickname list entry format (ADMTNICK) 164

non-IBM devices diagnosis 12

NULL function in TRCESTR statements 34

O

operating environment reporting 140

operating system problems 4

operators, use in TRCESTR statements 29

OS/2 Problem Report Form 122

OS/2.

finding level of code 107

output

incorrect 1

missing 2

unexpected 1

P

PARMSF function in TRCESTR statements 34

PARTIO function in TRCESTR statements 34

PCIA 9

PCIA (print control information area) 8

PCLK

See GDDM-PCLK

pclk command 98

pclkev command 98

PCSMON formatter program 104

performance problems, system 18

line time-outs 18

missing interrupt conditions 18

NCP slowdown condition 18

performance, reduced, reporting 139

Personal Communications/3270 debugging hints 104

Personal Communications/3270 tracing 104

personal computer system screens

dual 99

single 99

pitfalls during installation 12

POSTSCR

PostScript tracing function 60

POSTSCR function in TRCESTR statements 35

PostScript output

tracing 60

print control information area (PCIA) 8

PRINT function in TRCESTR statements 35

control string parameter 35

examples 36

remaining parameters 36

index

- print utilities, GDDM, problems with 4
- printers
 - diagnosis 9
 - diagnosis on 3268 9
 - diagnosis on 3287 8
 - diagnosis, general 7
- printing quality 13
- problem determination
 - GDDM-OS/2 Link 109
 - GDDM-PCLK 98
- problem report form 136
- problems 135, 142
 - after a GDDM installation 5
 - device checks 18
 - diagnosis of 1, 135
 - Graphics Control Program errors 20
 - installation 12
 - reporting to IBM 135, 142
 - describing the devices in use 141
 - describing the operating symptom 140
 - describing the symptom 138
 - SNA sense codes 19
 - system performance 18
 - unexpected message encountered 14
 - user abends 13
 - with application programs 3
 - with customizing workstations 4
 - with GDDM 5
 - with GDDM print utilities 4
 - with GDDM utility programs 3
 - with hardware 4
 - with operating system 4
- procedures, diagnostic 1
- PROG752 device check 19
- program check 19
 - reporting 138
- program failure, reporting 135
- publications
 - related 230

R

- RANGE function in TRCESTR statements 41
 - example 44
- RCP
 - codes, list of 203
- RCP code, use in TRCESTR statements 29
- record types, trace 43
- records, trace, format of 42
- reduced performance, reporting 139
- related publications 230
- Reporting Personal Communications/3270 Problems 105
- reporting problems to IBM 135, 138, 140
 - describing the devices in use 141
 - describing the operating environment 140

- reporting problems to IBM (*continued*)
 - describing the symptom 138
- reporting problems with network tools 135
- running GDDM-PCLK
 - with service functions 98

S

- sample trace statements 26
- save area chain location 75
- sense codes 2
 - (SNA) problems with 19
 - 1005 19
 - 800A 20
- service applied in the host for GDDM-PCLK 98
- service functions
 - GDDM-OS/2 Link 109
 - GDDM-PCLK 98
- service functions, GDDM-PCLK
 - panel 99
- Session n, GDDM-OS/2 Link 115
- SFAP (structured field and attribute processing) 7
- SNA 12
- SNA sense codes, problems with 19
- SPINIT, GDDM call for tracing 69, 70
- starting GDDM-PCLK
 - with service functions 98
- starting trace 22
- STGREP function in TRCESTR statements 37
- storage
 - address 86
 - display 86
- structured field and attribute processing (SFAP) 7
- SUBIN trace record 43, 44
 - in in-storage trace table 77
- SUBOUT trace record 43, 44
 - in in-storage trace table 77
- symbol sets 4
- symptom reporting 138
- syntax, trace-string 199
- SYSLEVEL command 107
- system abend, reporting 139
- system performance problems 18
 - line time-outs 18
 - missing interrupt conditions 18
 - NCP slowdown condition 18
- system, operating, problems with 4

T

- TDB (terminal descriptor block) 166
- terminal descriptor block (ADMTTDB) 166
- terminal request block (ADMTTRB) 168
- terminal services interface control area (ADMTTCA) 165

- TIME function in TRCESTR statements 37
- time-outs 16, 17, 18
- trace (communications manager),
 - GDDM-OS/2 Link 117
- trace control word 69
- trace explanation 43
- trace facility
 - GDDM-PCLK 99
- TRACE function in TRCESTR statements 37
- trace options windows (GDDM-OS/2 Link) 112
- trace points
 - (GDDM-PCLK) 99
 - GDDM-OS/2 Link 112
- trace record format in in-storage trace table 75
- trace record types 43
- trace statements
 - samples 26
- trace support, GDDM-OS/2 Link 109
- trace-string syntax 199
- trace, starting 22
- tracing 21, 91
 - abend dump 72
 - errors
 - GDDM-OS/2 Link 109
 - GDDM-PCLK 98
 - examples 44
 - CDPDS input 58
 - GDDM-REXX 93
 - GDDM-REXX/VM 92
 - HRIG output 59
 - multiple instances of GDDM 57
 - PostScript output 60
 - restriction by IF statement 46
 - using 5080IO function 55
 - using FREGS function 52
 - using FULLIO function 49
 - using GREGS function 52
 - using RANGE function 44
 - in a windowing environment 25
 - in an application program 25
 - in the TSO print utility 23
 - keywords, summary of 27
 - multiple instances of GDDM 25
 - output record format 42
 - TRCESTR statements 27
 - under CICS 23
 - under IMS 24
 - under TSO 22
 - under VM 23
 - under VSE/Batch 25
- tracing methods 69
- tracing MSL operations 87
- tracing output, GDDM-OS/2 Link 116
- trademarks xiv
- transmission buffer, trace output, GDDM-PCLK 102
- TRB (terminal request block) 168
- TRCESHR 27
 - default 26
- TRCESTR
 - 5080IO function 38
 - ABEND function 31
 - action functions
 - order of evaluation 31
 - summary 30
 - ALL function 38
 - CDPDS function 31
 - CGMREP function 32
 - CLEAR function 30
 - component function 39
 - compound statements 28
 - control function 42
 - COUNT function 39
 - DSOPEN function 32
 - ENTRY function 40
 - event functions summary
 - EXIT function 40
 - FLOW function 32
 - FORCE function 30
 - FREGS function 33
 - FULLIO function 33
 - FULLTCA function 33
 - functions available 30
 - GREGS function 33
 - HRIG function 33
 - introduction to 27
 - keyword, summary of functions 22
 - level function 40
 - LIMIT function 40
 - LIST function 34
 - multiple actions in one statement 28
 - NAME function 41
 - nesting IF statements 28
 - NULL function 34
 - PARMSF function 34
 - PARTIO function 34
 - POSTSCR function 35
 - PRINT function 35
 - RANGE function 41
 - statements, coding 27
 - statements, detailed syntax of 199
 - STGREP function 37
 - TIME function 37
 - TRACE function 37
 - use of % in statements 29
 - use of double quotation marks in statements 28, 37, 41
 - use of floating-point registers (FR) 29
 - use of general-purpose registers (GR) 29
 - use of operators 29
 - use of RCP code 29

index

TRCEWID keyword 42
TRTABLE 27
TSO
 tracing in the print utility 23
 tracing under 22
TSOTRCE 27

U

unexpected output, reporting 138
update window, GDDM-OS/2 Link 112
user abends
 code 1201 13
 code 2053 13
 code 2054 13
 code 2201 13
 code G201 13
user session logoff 14, 16, 17, 18
utility programs, GDDM, problems with 3

V

VM display communication block 148
VM, tracing under 23
VSE/Batch, tracing in 25

W

WACK support 11
wait, reporting 139
window procedure, GDDM-OS/2 Link 114
windowing environment, tracing in 25
workstation customization, problems 4

X

XHGI component, trace output, GDDM-PCLK 102

Sending your comments to IBM

GDDM

Diagnosis

SC33-0870-01

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book and the way in which the information is presented.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, use the Readers' Comment Form
- By fax:
 - From outside the U.K., after your international access code use 44 1962 870229
 - From within the U.K., use 01962 870229
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink: WINVMD(IDRCF)
 - Internet: idrcf@winvmd.vnet.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

Readers' Comments

GDDM

Diagnosis

SC33-0870-01

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Telephone

Email

You can send your comments POST FREE on this form from any one of these countries:

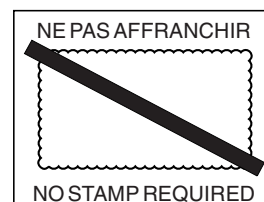
Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

2 Fold along this line

By air mail
Par avion

IBRS/CCRI NUMBER: PHQ - D/1348/SO



REPONSE PAYEE
GRANDE-BRETAGNE

IBM United Kingdom Laboratories
Information Development Department (MP095)
Hursley Park,
WINCHESTER, Hants
SO21 2ZZ United Kingdom

3 Fold along this line

From: Name _____
Company or Organization _____
Address _____

EMAIL _____
Telephone _____

4 Fasten here with adhesive tape



1 Cut along this line

1 Cut along this line



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-0870-01

